Quantum

Configuring and Maintaining a Storage Array Using CLI

Quantum StorNext QS/QD SANtricity Storage Manager 11.20

6-68264-01 Rev A

StorNext Q-Series Storage Configuring and Maintaining a Storage Array Using CLI, 6-68264-01 Rev A, May 2015, Product of USA.

Table of Contents

Deciding Whether to Use This Guide

This guide describes how to configure and maintain a QS1200, QS2400, QD6000, and QD7000 base system storage array using the command line interface (CLI). Using the CLI, you can run commands from an operating system prompt or from a script engine window in the SANtricity Storage Manager Enterprise Management Window (EMW).

Use this guide if you want to accomplish these goals:

- Learn about the CLI
- Learn about script commands
- Learn configuration concepts
- Configure a storage array using the CLI
- Use any of the following features via the CLI:
  - Snapshot (legacy) feature
  - Snapshot Image feature
  - Asynchronous Mirroring feature
  - Synchronous Mirroring feature
  - Volume Copy feature
  - SSD Cache feature
- Maintain a storage array using the CLI
- See example script files
- Use the Asynchronous Write Mode Mirror Utility via the CLI
- Perform a Simplex-to-Duplex conversion using the CLI

This guide is based on the following assumptions:

- Your storage system has been successfully installed.
- SANtricity Storage Manager has been successfully installed.

Where to Find the Latest Information About the Product

You can find information about the latest version of the product, including new features and fixed issues, and a link to the latest documentation at the following address: Quantum.com

How to send your comments

You can help us to improve the quality of our documentation by sending us your feedback.

Your feedback is important in helping us to provide the most accurate and high-quality information. If you have suggestions for improving this document, send us your comments to: Quantum.com. To help us direct your comments to the correct division, include in the subject line the product name, version, and operating system.

For further assistance, or if training is desired, contact the Quantum Customer Support Center:
United States
1-800-284-5101 (toll free)
+1-720-249-5700
EMEA
+800-7826-8888 (toll free)
+49-6131-3241-1164
APAC
+800-7826-8887 (toll free)
+603-7953-3010

Chapter 1 - About the Command Line Interface

The command line interface (CLI) is a software application that provides a way for installers, developers, and engineers to configure and monitor storage arrays. Using the CLI, you can run commands from an operating system prompt, such as the DOS `C:` prompt, a Linux operating system path, or a Solaris operating system 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. For example, you can run script files when you want to install the same configuration on several storage arrays. The CLI enables you to load a script file from a disk and run the script file. The CLI provides a way to run storage management commands on more than one network storage array. You can use the CLI both in installation sites and in development environments.

The CLI gives you direct access to a script engine that is a utility in the SANtricity Storage Manager software (also referred to as the storage management software). The script engine runs commands that configure and manage the 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 by using the Enterprise Management Window in the storage management software. If you access the script engine by 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. By using the CLI, you can run commands on more than one storage array from a single command line.

You can use the command line interface to perform these actions:

- 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
- Show configuration information about the network storage arrays
- Add storage arrays to and remove storage arrays from the management domain
- Perform automatic discovery of all of the storage arrays that are 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
- Show the alert notification settings for storage arrays that are currently configured in the Enterprise Management Window
- Direct the output to a standard command line display or to a named file

Structure of a CLI Command

The CLI commands are in the form of a command wrapper and elements embedded into the wrapper. A CLI command consists of these elements:

- A command wrapper identified by the term **`SMcli`**
- The storage array identifier

- Terminals that define the operation to be performed
- Script commands

The CLI command wrapper is a shell that identifies storage array controllers, embeds operational terminals, embeds script commands, and passes these values to the script engine.

All CLI commands have the following structure:

`SMcli` *`storageArray terminal script-commands`*`;`

- **`SMcli`** invokes the command line interface.
- *`storageArray`* is the name or the IP address of the storage array.
- *`terminal`* are CLI values that define the environment and the purpose for the command.
- *`script-commands`* are one or more script commands or the name of a script file that contains script commands. (The script commands configure and manage the storage array.)

If you enter an incomplete or inaccurate **`SMcli`** string that does not have the correct syntax, parameter names, options, or terminals, the script engine returns usage information.

Interactive Mode

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 lets you run individual commands without prefixing the commands with **`SMcli`**.

In interactive mode, you can enter a single command, view 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.

To end an interactive mode session, type the operating system-specific command for terminating a program, such as **Control-C** on the UNIX operating system or the Windows operating system. Typing the termination command (**Control-C**) while in interactive mode turns off interactive mode and returns operation of the command prompt to an input mode that requires you to type the complete **`SMcli`** string.

CLI Command Wrapper Syntax

General syntax forms of the CLI command wrappers are listed in this section. The general syntax forms show the terminals and the parameters that are used in each command wrapper. The conventions used in the CLI command wrapper syntax are listed in the following table.

Convention	Definition
`a | b`	Alternative ("a" or "b")
`italicized-words`	A terminal that needs user input to fulfill a parameter (a response to a variable)
`[ ... ]` (square brackets)	Zero or one occurrence (square brackets are also used as a delimiter for some command parameters)

Convention	Definition
{ ... } (curly braces)	Zero or more occurrences
(a | b | c)	Choose only one of the alternatives
bold	A terminal that needs a command parameter entered to start an action

ATTENTION To run all of the CLI command you must have administrator privileges. Some CLI commands will run without administrator privileges. Many commands, however, will not run. If the CLI command does not run because you do not have correct privileges, the CLI returns an exit code of 12.

```
SMcli host-name-or-IP-address [host-name-or-IP-address]
[-c "command; {command2};"]
[-n storage-system-name | -w wwID]
[-ooutputfile] [-ppassword] [-R (admin | monitor)] [-e] [-S] [-quick]

SMcli host-name-or-IP-address [hostname-or-IP-address]
[-fscriptfile]
[-nstorage-system-name | -wwwID]
[-ooutputfile] [-ppassword] [-R (admin | monitor)] [-e] [-S] [-quick]

SMcli (-n storage-system-name | -wwwID)
[-c "command; {command2};"]
[-ooutputfile] [-ppassword] [-R (admin | monitor)] [-e] [-S] [-quick]

SMcli (-nstorage-system-name -wwwID)
[-fscriptfile]
[-ooutputfile] [-R (admin | monitor)] [-ppassword] [-e] [-S] [-quick]

SMcli -aemail:email-address [host-name-or-IP-address1
[host-name-or-IP-address2]]
[-nstorage-system-name | -wwwID | -hhost-name]
[-Iinformation-to-include] [-qfrequency] [-S]

SMcli -xemail:email-address [host-name-or-IP-address1
[host-name-or-IP-address2]]
[-nstorage-system-name | -wwwID | -hhost-name] [-S]

SMcli (-a | -x) trap:community, host-name-or-IP-address
[host-name-or-IP-address1 [host-name-or-IP-address2]]
[-nstorage-system-name | -wwwID | -hhost-name] [-S]

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

SMcli -mhost-name-or-IP-address-Femail-address
[-gcontactInfoFile] [-S]

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

SMcli -X (-nstorage-system-name | -wwwID | -hhost-name)

SMcli -?
```

Command Line Terminals

Terminal	Definition
host-name-or-IP-address	Specifies either the host name or the Internet Protocol (IP) address (*xxx.xxx.xxx.xxx*) of an in-band managed storage array or an out-of-band managed storage array. ■ If you are managing a storage array by using a host through in-band storage management, you must use the **-n** terminal or the **-w** terminal if more than one storage array is connected to the host. ■ If you are managing a storage array by using out-of-band storage management through the Ethernet connection on each controller, you must specify the *host-name-or-IP-address* of the controllers. ■ If you have previously configured a storage array in the Enterprise Management Window, you can specify the storage array by its user-supplied name by using the **-n** terminal. ■ If you have previously configured a storage array in the Enterprise Management Window, you can specify the storage array by its World Wide Identifier (WWID) by using the **-w** terminal.
-A	Adds a storage array to the configuration file. If you do not follow the **-A** terminal with a *host-name-or-IP-address*, auto-discovery scans the local subnet for storage arrays.
-a	Adds a Simple Network Management Protocol (SNMP) trap destination or an email address alert destination. ■ When you add an SNMP trap destination, the SNMP community is automatically defined as the community name for the trap, and the `host` is the IP address or Domain Name Server (DNS) host name of the system to which the trap should be sent. ■ When you add an email address for an alert destination, the `email-address` is the email address to which you want the alert message to be sent.
-c	Indicates that you are entering one or more script commands to run on the specified storage array. End each command with a semicolon (**;**). You cannot place more than one **-c** terminal on the same command line. You can include more than one script command after the **-c** terminal.
-d	Shows the contents of the script configuration file. The file content has this format: *storage-system-name host-name1 host-name2*
-e	Runs the commands without performing a syntax check first.
-F (uppercase)	Specifies the email address from which all alerts will be sent.
-f (lowercase)	Specifies a file name that contains script commands that you want to run on the specified storage array. The **-f** terminal is similar to the **-c** terminal in that both terminals are intended for running script commands. The **-c** terminal runs individual script commands. The **-f** terminal runs a file of script commands. By default, any errors that are encountered when running the script commands in a file are ignored, and the file continues to run. To override this behavior, use the **set session errorAction=stop** command in the script file.
-g	Specifies an ASCII file that contains email sender contact information that will be included in all email alert notifications. The CLI assumes that the ASCII file is text only, without delimiters or any expected format. Do not use the **-g** terminal if a `userdata.txt` file exists.

Terminal	Definition
-h	Specifies the host name that is running the SNMP agent to which the storage array is connected. Use the **-h** terminal with these terminals: ■ **-a** ■ **-x**
-I (uppercase)	Specifies the type of information to be included in the email alert notifications. You can select these values: ■ **eventOnly** – Only the event information is included in the email. ■ **profile** – The event and array profile information is included in the email. You can specify the frequency for the email deliveries using the **-q** terminal.
-i (lowercase)	Shows the IP address of the known storage arrays. Use the **-i** terminal with the **-d** terminal. The file content has this format: *storage-system-name IP-address1 IPaddress2*
-m	Specifies the host name or the IP address of the email server from which email alert notifications are sent.
-n	Specifies 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*. If you are using the in-band method for managing the storage array, you must use the **-n** terminal if more than one storage array is connected to the host at the specified address. The storage array name is required when the *host-name-or-IP-address* is not used. The name of the storage array that is configured for use in the Enterprise Management Window (that is, the name is listed in the configuration file) must not be a duplicate name of any other configured storage array.
-o	Specifies a file name for all output text that is a result of running the script commands. Use the **-o** terminal with these terminals: ■ **-c** ■ **-f** If you do not specify an output file, the output text goes to standard output (stdout). All output from commands that are not script commands is sent to stdout, regardless of whether this terminal is set.
-p	Defines the password for the storage array on which you want to run commands. A password is not necessary if a password has not been set on the storage array. If you set a monitor password for the storage array, the use of the -p parameter is mandatory. Users cannot run any of the non-destructive commands such as the **show** commands.

Terminal	Definition
-q	Specifies the frequency that you want to receive event notifications and the type of information returned in the event notifications. An email alert notification containing at least the basic event information is always generated for every critical event. These values are valid for the **-q** terminal: ■ **everyEvent** – Information is returned with every email alert notification. ■ **2** – Information is returned no more than once every two hours. ■ **4** – Information is returned no more than once every four hours. ■ **8** – Information is returned no more than once every eight hours. ■ **12** – Information is returned no more than once every 12 hours. ■ **24** – Information is returned no more than once every 24 hours. Using the **-I** terminal you can specify the type of information in the email alert notifications. ■ If you set the **-I** terminal to **eventOnly**, the only valid value for the **-q** terminal is **everyEvent**. ■ If you set the **-I** terminal to either the **profile** value or the **supportBundle** value, this information is included with the emails with the frequency specified by the **-q** terminal.
-quick	Reduces the amount of time that is required to run a single-line operation. An example of a single-line operation is the **recreate snapshot volume** command. This terminal reduces time by not running background processes for the duration of the command. Do not use this terminal for operations that involve more than one single-line operation. Extensive use of this command can overrun the controller with more commands than the controller can process, which causes operational failure. Also, status updates and configuration updates that are collected usually from background processes will not be available to the CLI. This terminal causes operations that depend on background information to fail.
-R (uppercase)	Defines the user role for the password. The roles can be either: ■ `admin` – The user has privilege to change the storage array configuration. ■ `monitor` – The user has privilege to view the storage array configuration, but cannot make changes. The `-R` parameter is valid only when used with the `-p` parameter, which specifies that you define a password for a storage array. The `-R` parameter is required only if the dual password feature is enabled on the storage array. The `-R` parameter is not necessary under these conditions: ■ The dual password feature is not enabled on the storage array. ■ Only one admin role is set and the monitor role is not set for the storage array.

Terminal	Definition
-S (uppercase)	Suppresses informational messages describing the command progress that appear when you run script commands. (Suppressing informational messages is also called silent mode.) This terminal suppresses these messages: ■ Performing syntax check ■ Syntax check complete ■ Executing script ■ Script execution complete ■ SMcli completed successfully
-s (lowercase)	Shows the alert settings in the configuration file when used with the **-d** terminal.
-v	Shows the current global status of the known devices in a configuration file when used with the **-d** terminal.
-w	Specifies the WWID of the storage array. This terminal is an alternate to the **-n** terminal. Use the **-w** terminal with the **-d** terminal to show the WWIDs of the known storage arrays. The file content has this format: *storage-system-name world-wide-ID IP-address1 IP-address2*
-X (uppercase)	Deletes a storage array from a configuration.
-x (lowercase)	Removes an SNMP trap destination or an email address alert destination. The *community* is the SNMP community name for the trap, and the *host* is the IP address or DNS host name of the system to which you want the trap sent.
-?	Shows usage information about the CLI commands.

AutoSupport Bundle Collection Commands

AutoSupport (ASUP) is a feature that enables storage arrays to automatically collect support data into a customer support bundle and send the data to Technical Support. Technical Support can then perform remote troubleshooting and problem analysis with the storage management software. ASUP collects support data to report configuration, subsystem status, and exceptions in near-real time. ASUP messages typically include a collection of system log files, configuration data (formatted XML and unstructured command output), state data (subsystem up/down, capacity used), performance metrics, and system inventory data. All of the data gathered is collected into a single compressed archive file format (7z).

With the implementation of ASUP, users have two possible methods for collecting support data in a storage array:

- ASUP collection

  Data is automatically collected and sent to Technical Support.
- Legacy support bundle collection

  Collection of legacy support bundle data is configured by the user at intervals scheduled by the user. Users can then manually send the support bundles to Technical Support.

ASUP operations and legacy support bundle operations are mutually exclusive on a given storage array. When you turn on ASUP you automatically disable legacy support bundle collection. If you want to run legacy support bundle collection, you must turn off ASUP.

In the CLI, ASUP is a nonconfigurable, "set it and forget it" feature. Using the CLI commands, you can only turn on or turn off ASUP. Once turned on, ASUP automatically reports configuration, subsystem status, and exceptions in near-

real time. Because ASUP speeds up troubleshooting and problem analysis, ASUP is the preferred data collection method to use if available on the storage array.

ASUP Messages

ASUP provides these types of messages:

- Event:
  - Sent when a support event occurs on the managed storage array.
  - Includes system configuration and diagnostic information.
  - Includes minimal extent of system configuration information.
- Daily:
  - Sent at midnight, local time of the host.
  - Provides a current set of system event logs and performance data.
  - Places less burden on payload and transmission on the messages originating from Event ASUP messages.
- Weekly:
  - Sent once every week at times that do not impact storage array operations.
  - Includes configuration and system state information.

The storage management software automatically assigns the schedule for each storage array it has discovered.

The storage array uses the internet to send ASUP messages to the ASUP backend. The ASUP backend provides near-real time access to the messages by Technical Support. ASUP requires compliance to the following transport protocol-specific requirements:

- HTTP or HTTPS upload
- SMTP notifications

ASUP Commands

The CLI ASUP commands in the following table turn on or turn off the ASUP feature for either all of the storage arrays managed at the Enterprise Management Window (EMW) level or for a specific storage array.

`SMcli enable autoSupportFeature`	Turns on the ASUP feature at the EMW level
`SMcli disable autoSupportFeature`	Turns off the ASUP feature at the EMW level
`set storageArray autoSupportFeature enable`	Turns on the ASUP feature for a specific storage array
`set storageArray autoSupportFeature disable`	Turns off the ASUP feature for a specific storage array

The two SMcli commands run at the EMW level. All of the storage arrays being managed that are ASUP capable can be enabled or disabled using the commands. As shown in the table, these are the complete commands.

The two **`set`** commands are script commands that you can use to turn on or turn off ASUP for individual storage arrays. You can run these commands from the script editor in the storage management software GUI, a script file, or from the command line if you use a CLI wrapper as shown in the following example:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “set storageArray autoSupportFeature enable;”
```

ASUP Log

The ASUP log file has a detailed list of events encountered during delivery of the ASUP messages. The ASUP log provides information about status, history of transmission activity, and any errors encountered during delivery of the ASUP messages. The log file is available for all ASUP-enabled storage arrays.

The archived log filename is `ASUPMessages.n`, where `n` is an integer from 1 to 5. The log file is located in the `ASUPLog` directory. As the current log file reaches a size limit of 200 KB, the current log file is archived and a new log file is created.

Disable AutoSupport at the EMW Level SMcli Version

NOTE This command is an SMcli command, not a script command. You must run this command from a command line. You cannot run this command from the script editor in the storage management software.

This command turns off the AutoSupport (ASUP) bundle collection feature for all managed storage arrays.

Syntax

```
SMcli disable autoSupportFeature
```

Parameters

None.

Minimum Firmware Level

7.86

Enable AutoSupport at the EMW Level SMcli Version

NOTE This command is an SMcli command, not a script command. You must run this command from a command line. You cannot run this command from the script editor in the storage management software.

This command turns on the AutoSupport (ASUP) bundle collection feature for all managed storage arrays and makes it possible to transmit the bundle to a predesignated technical support site. After you enable the ASUP feature, any ASUP-capable storage array is automatically prepared to collect and send support-related data to Technical Support. The data can then be used for remote troubleshooting and problem analysis.

Syntax

```
SMcli enable autoSupportFeature
```

Parameters

None.

Minimum Firmware Level

7.86

Set Storage Array AutoSupport Bundle Disable

This command turns off the AutoSupport (ASUP) bundle collection and transmission for the storage array. You can run this version of the command from the script editor or in a script file.

Syntax

```
set storageArray autoSupportFeature disable
```

Parameters

None.

Minimum Firmware Level

7.86

Set Storage Array AutoSupport Bundle Enable

This command turns on the AutoSupport (ASUP) bundle collection and transmission for the storage array.

Syntax

```
set storageArray autoSupportFeature enable
```

Parameters

None.

Minimum Firmware Level

7.86

Test AutoSupport Configuration

NOTE This command is an SMcli command, not a script command. You must run this command from a command line. You cannot run this command from the script editor in the storage management software

This command validates the current auto support configuration by sending a test ASUP bundle and reports the result of the test transmission.

Syntax

```
SMcli -supportBundle testConfig
```

Parameters

None

Minimum Firmware Level

7.86

Naming Conventions

- Names can have a maximum of 30 characters.
- You can use any combination of alphanumeric characters, hyphens, and underscores for the names of the following components:
  - Storage arrays
  - Host groups
  - Hosts
  - Volume groups
  - Volumes
  - HBA host ports
- You must use unique names. If you do not use unique names, the controller firmware returns an error.
- If the name contains more than one word, hyphens, or underscores, enclose the name in double quotation marks (“ ”). In some usages, you must also surround the name with square brackets ([ ]). The description of each parameter indicates whether you need to enclose a parameter in double quotation marks, square brackets, or both.
- The name character string cannot contain a new line.
- On Windows operating systems, you must enclose the name between two backslashes (\\) in addition to other delimiters. For example, the following name is used in a command that runs under a Windows operating system:

  ```
  [\“Engineering\”]
  ```

- For a UNIX operating system and, when used in a script file, the name appears as in the following example:

  ```
  [“Engineering”]
  ```

- When you enter a World Wide Identifier (WWID) of an HBA host port, some usages require that you surround the WWID with double quotation marks. In other uses, you must surround the WWID with angle brackets (<>). The description of the WWID parameter indicates whether you need to enclose the WWID in double quotation marks or angle brackets.

Entering Numerical Names

When the storage management software automatically configures a storage array, the storage management software assigns names that consist of numerical characters. Names that consist only of numerical characters are valid names. Numerical character names, however, must be treated differently than names that start with alphabetic characters.

- Names that are only numbers, such as 1 or 2
- Names that start with a number, such as 1Disk or 32Volume
- `[\“1\”]`
- `[\“1Disk\”]`

NOTE If you have any doubt as to the validity of a name, use both double quotation marks and square brackets. Using both makes sure that the name will work, but will not cause any processing issues.

Formatting CLI Commands

Double quotation marks (" ") that are used as part of a name or label require special consideration when you run the CLI commands and the script commands on a Microsoft Windows operating system.

When double quotation marks (" ") are part of a name or value, you must insert a backslash (\) before each double quotation mark character. For example:

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

In this example, "Engineering" is the storage array name. A second example is:

```
-n \"My\"_Array
```

In this example, "My"_Array is the name of the storage array.

You cannot use double quotation marks (" ") 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 the Linux operating system and the Solaris operating system, the delimiters around names or labels are single quotation marks (' '). The UNIX versions of the previous examples are as follows:

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

```
-n "My"_Array
```

In a Windows operating system, if you do not use double quotation marks (" ") around a name, you must insert a caret (^) before each special script character. Special characters are ^, | , <, and >.

Insert a caret before each special script character when used with the terminals **-n**, **-o**, **-f**, and **-p**. For example, to specify storage array CLI>CLIENT, enter this string:

```
-n CLI^>CLIENT
```

Insert one caret (^) before each special script character when used within a string literal in a script command. For example, to change the name of a storage array to FINANCE_|_PAYROLL, enter the following string:

```
-c "set storageArray userLabel=\"FINANCE_^|_PAYROLL\";"
```

Formatting Rules for Script Commands

Syntax unique to a specific script command is explained in the Notes section at the end of each script command description.

Case sensitivity – The script commands are not case sensitive. You can type the script commands in lowercase, uppercase, 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.)

Spaces – You must enter spaces in the script commands as they are shown in the command descriptions.

Square brackets – Square brackets are used in two ways:

- As part of the command syntax.
- To indicate that the parameters are optional. The description of each parameter tells you if you need to enclose a parameter value in square brackets.

Parentheses – 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 do not include parentheses in a script 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 values and slot ID values. The description of each parameter tells you if you need to enclose a parameter value in parentheses.

Vertical bars – Vertical bars in a script command indicate "or" and separate the valid values for the parameter. For example, the syntax for the `raidLevel` parameter in the command description appears as follows:

```
raidLevel=(0 | 1 | 3 | 5 | 6)
```

To use the `raidLevel` parameter to set RAID level 5, enter this value:

```
raidLevel=5
```

Drive locations – The CLI commands that identify drive locations support both high-capacity drive trays and low-capacity drive trays. A high-capacity drive tray has drawers that hold the drives. The drawers slide out of the drive tray to provide access to the drives. A low-capacity drive tray does not have drawers. For a high-capacity drive tray, you must specify the identifier (ID) of the drive tray, the ID of the drawer, and the ID of the slot in which a drive resides. For a low-capacity drive tray, you need only specify the ID of the drive tray and the ID of the slot in which a drive resides. For a low-capacity drive tray, an alternative method for identifying a location for a drive is to specify the ID of the drive tray, set the ID of the drawer to 0, and specify the ID of the slot in which a drive resides. Separate the ID values with a comma. If you enter more than one set of ID values, separate each set of values with 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)
```

or, for a high-capacity drive tray, this example:

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

Italicized terms – Italicized terms in the command indicate a value or information that you need to provide. For example, when you encounter the italicized term:

numberOfDrives

Replace the italicized term with a value for the number of drives that you want to include with the script command.

Semicolon – Script commands must end with a semicolon (`;`). You can enter more than one script command on the command line or in a script file. For example, a semicolon is used to separate each script command in the following script file.

```
create volume drives=(0,2 0,3 1,4 1,5 2,6 2,7) raidLevel=5
userLabel="v1" capacity=2gb owner=a;
```

```
create volume volumeGroup=2 userLabel=”v2” capacity=1gb owner=b;
create volume volumeGroup=2 userLabel=”v3” capacity=1gb owner=a;

create volume drives=(0,4 0,5 1,6 1,7 2,8 2,9) raidLevel=5
userLabel=”v4” capacity=2gb owner=b;
create volume volumeGroup=3 userLabel=”v5” capacity=1gb owner=a;
create volume volumeGroup=3 userLabel=”v6” capacity=1gb owner=b;
```

Formatting CLI Commands in Windows PowerShell

The Windows PowerShell is an interactive and scripting shell that provides access to command-line tools. The Windows PowerShell improves upon the Windows Command Prompt with a more robust set of commands and scripting capability. You can run all of the CLI and script commands in the Windows PowerShell; however, the Windows PowerShell has some unique formatting requirements. The requirements are these:

- Begin all SMcli commands with a period and a forward slash (./)
- SMcli wrapper must be identified as an executable command with the .exe extension (SMcli.exe)
- Enclose the script command in single quotation marks (’ ’)
- Double quotation marks that are part of a name, file path, or value must have a backslash before each double quotation mark character (\")

Following is an example of a CLI command to create a storage array name in the Windows Powershell. Note the use of the single quotation marks as delimiters for the script command and the backslash double quotation marks around the storage array name, identified as `userLabel` in the command syntax.

```
PS C:\...\StorageManager\client> ./SMcli.exe 123.45.67.88 123.45.67.89
-c ’set storageArray userLabel=\"Engineering\";’
```

Following is an example of a CLI command to enable a premium feature in the Windows Powershell. Note the use of the backslash double quotation marks before the file path to the premium feature key.

```
PS C:\...\StorageManager\client> ./SMcli.exe 123.45.67.88 123.45.67.89
-c ’enable storageArray feature file=\"C:\licenseKey.bin\";’
```

In the previous examples both upper case letters and lower case letters are used. This use is to help make clear how the commands are formatted. The Windows PowerShell is, however, not case sensitive and does not require the use specific cases.

Usage Examples

The following examples show how to enter CLI commands on a command line. The examples show the syntax, the form, and, in some examples, script commands. Examples are shown for both the Windows operating system and the UNIX operating system. Note that the usage for the **-c** terminal varies depending on your operating system. On Windows operating systems, enclose the script command following the **-c** terminal in double quotation marks (" "). On UNIX operating systems, enclose the script command following the **-c** terminal in single quotation marks (' '). For descriptions of the script commands used in these examples, refer to the *Command Line Interface and Script Commands for Version 10.75*.

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 is 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";'
```

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 is 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 [\"Stocks_^<_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 ["Stocks_<Bonds"];
create volume driveCount[3] RAIDLEVEL=3 capacity=10GB userLabel="Finance";
show storageArray healthStatus;'
```

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

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

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 MyArray. Output, as a result of commands in the script file, goes 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
```

This example shows how to show all of the 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`** terminal to the command.

```
SMcli -d -i
```

Exit Status

This table lists the exit statuses that might be returned and the meaning of each status.

Status Value	Meaning
0	The command terminated without an error.
1	The command terminated with an error. Information about the error also appears.
2	The script file does not exist.
3	An error occurred while opening an output file.
4	A storage array was not at the specified address.
5	Addresses specify different storage arrays.
6	A storage array name does not exist for the host agent that is connected.
7	The storage array name was not at the specified address.
8	The storage array name was not unique.
9	The storage array name was not in the configuration file.
10	A management class does not exist for the storage array.
11	A storage array was not found in the configuration file.
12	An internal error occurred. This exit status indicates that you do not have privileges for running a CLI command from the command line. You must have administrator privileges to run all of the CLI commands from a command line.
13	Invalid script syntax was found.
14	The controller was unable to communicate with the storage array.
15	A duplicate argument was entered.
16	An execution error occurred.
17	A host was not at the specified address.
18	The WWID was not in the configuration file.
19	The WWID was not at the address.
20	An unknown IP address was specified.
21	The Event Monitor configuration file was corrupted.
22	The storage array was unable to communicate with the Event Monitor.
23	The controller was unable to write alert settings.
24	The wrong organizer node was specified.
25	The command was not available.
26	The device was not in the configuration file.
27	An error occurred while updating the configuration file.
28	An unknown host error occurred.
29	The sender contact information file was not found.
30	The sender contact information file could not be read.
31	The `userdata.txt` file exists.
32	An invalid **`-I`** value in the email alert notification was specified.
33	An invalid **`-f`** value in the email alert notification was specified.
34	The **`-r`** option is not supported anymore.
35	Invalid alert severity specified.

Status Value	Meaning
36	The operation needs either the Administrator or Monitor password to be set.
37	The operation cannot be completed because an invalid Monitor password was entered.
38	The operation cannot be completed because an invalid Administrator password was entered.
39	The password provided is exceeding the character limit.
40	The **-R** monitor is not supported for this array. Use a valid role and retry the operation.
42	Host address or mail server address is incorrect.

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 command line interface (CLI) command wrappers. You can enter individual script commands, or you can run a file of script commands. When you enter an individual script command, you embed the script command in a CLI command wrapper. When you run a file of script commands, you embed the file name in the CLI command wrapper. The script commands are processed by a script engine that performs the following functions:

- Verifies the 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 storage array controllers run the script commands.

The script engine and the script commands support the storage array configuration and management operations that are listed in the following table.

Table 1. Configuration and Management Operations

Operation	Activities
General storage array configuration	Resetting a configuration to defaults, labeling, checking the health status, setting the time of day, clearing the Event Log, and setting the media scan rate
Volume configuration and volume group configuration	Creating, deleting, and setting the reconstruction priority control, labeling, setting drive composition when creating volumes, setting the segment size, and setting the media scan control
Drive configuration	Assigning hot spares
Controller configuration	Defining volume ownership, changing mode settings, defining network settings, and setting host channel IDs
Firmware management	Downloading controller firmware, the environmental services module (ESM) firmware, and the drive firmware
NVSRAM configuration	Downloading and modifying the user configuration region at the bit level and the byte level, showing nonvolatile static random access memory (NVSRAM) values
Cache configuration	Controlling all cache parameters, both at the storage array level and the individual volume level
Product identification	Retrieving the tray profile display data
Battery management	Setting the battery installation date

Structure of a Script Command

All script commands have the following structure:

command operand-data (*statement-data*)

- *command* identifies the action to be performed.

- *operand-data* represents the objects associated with a storage array that you want to configure or manage.
- *statement-data* provides the information needed to perform the command.

The syntax for *operand-data* has the following structure:

```
(object-type | all object-types | [qualifier]
(object-type [identifier] (object-type [identifier] | object-types [identifier-list]))
```

An object can be identified in four ways:

- Object type – Use when the command is not referencing a specific object.
- `all` parameter prefix – Use when the command is referencing all of the objects of the specified type in the storage array (for example, `allVolumes`).
- Square brackets – Use when performing a command on a specific object to identify the object (for example, `volume [engineering]`).
- A list of identifiers – Use to specify a subset of objects. Enclose the object identifiers in square brackets (for example, `volumes [sales engineering marketing]`).

A qualifier is required if you want to include additional information to describe the objects.

The object type and the identifiers that are associated with each object type are listed in this table.

Table 2. Script Command Object Type Identifiers

Object Type	Identifier
`controller`	a or b
`drive`	Tray ID, Drawer ID, and slot ID
`replacementDrive`	Tray ID, Drawer ID, and slot ID
`driveChannel`	Drive channel identifier
`host`	User label
`hostChannel`	Host channel identifier
`hostGroup`	User label
`hostPort`	User label
`iscsiInitiator`	User label or iSCSI Qualified Name (IQN)
`iscsiTarget`	User label or IQN
`snapshot (legacy)`	Volume user label
`storageArray`	Not applicable
`tray`	Tray ID
`volume`	Volume user label or volume World Wide Identifier (WWID) (**`set`** command only)
`volumeCopy`	Target volume user label and, optionally, the source volume user label
`volumeGroup`	User label Valid characters are alphanumeric, a hyphen, and an underscore.

Statement data is in the form of:

- Parameter = value (such as `raidLevel=5`)
- Parameter-name (such as `batteryInstallDate`)

- Operation-name (such as `redundancyCheck`)

A user-defined entry (such as user label) is called a variable. In the syntax, it is shown in italic (such as *`trayID`* or *`volumeGroupName`*).

Synopsis of the Script Commands

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 definitions, 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 configure or maintain a storage array. The following table lists the general form of the script commands and a definition of each command.

Table 3. General Form of the Script Commands

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

Syntax	Description
`recover` *object* {*statement-data*}	Re-creates an object from saved configuration data and the statement parameters. (This command is similar to the **`create`** command.)
`recreate` *object* {*statement-data*}	Restarts a snapshot (legacy) operation by using an existing snapshot (legacy) volume. You can change the parameters before the operation is restarted.
`remove` *object* {*statement-data*}	Removes a relationship between objects.
`repair` *object* {*statement-data*}	Repairs errors found by the **`check`** command.
`replace` *object* {*statement-data*}	The specified object replaces an existing object in the storage array.
`reset` *object* {*statement-data*}	Returns the hardware or an object to an initial state.
`resume` *object*	Starts a suspended operation. The operation starts where it left off when it was suspended.
`revive` *object*	Forces the object from the Failed state to the Optimal state. Use this command only as part of an error recovery procedure.
`save` *object* {*statement-data*}	Writes information about the object to a file.
`set` *object* {*statement-data*}	Changes object attributes. All changes are completed when the command returns.
`show` *object* {*statement-data*}	Shows information about the object.
`start` *object* {*statement-data*}	Starts an asynchronous operation. You can stop some operations after they have started. You can query the progress of some operations.
`stop` *object* {*statement-data*}	Stops an asynchronous operation.
`suspend` *object* {*statement-data*}	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 parameters and options that you can use in the script commands. Table 4 lists the recurring syntax parameters and the values that you can use with the recurring syntax parameters. The conventions used in the recurring syntax elements are listed in the following table.

Convention	Definition
`a | b`	Alternative ("a" or "b")
`italicized-words`	A terminal that needs user input to fulfill a parameter (a response to a variable)
`[ ... ]` (square brackets)	Zero or one occurrence (square brackets are also used as a delimiter for some command parameters)

Convention	Definition
{ ... } (curly braces)	Zero or more occurrences
(a | b | c)	Choose only one of the alternatives
bold	A terminal that needs a command parameter entered to start an action

Table 4. Recurring Syntax Elements

Recurring Syntax	Syntax Value
raid-level	(0 | 1 | 3 | 5 | 6)
repository-raid-level	(1 | 3 | 5 | 6)
capacity-spec	*integer-literal* [KB | MB | GB | TB | Bytes]
segment-size-spec	*integer-literal*
boolean	(TRUE | FALSE)
user-label	*string-literal* Valid characters are alphanumeric, the dash, and the underscore.
user-label-list	*user-label* {*user-label*}
create-raid-vol-attr- value-list	*create-raid-volume-attribute-value-pair* {*create-raid-volume-attribute-value-pair*}
create-raid-volume-attribute-value-pair	capacity=*capacity-spec* | owner=(a | b) | cacheReadPrefetch=(TRUE | FALSE) | segmentSize=*integer-literal* | usageHint=*usage-hint-spec*
noncontroller-trayID	(0-99)
slotID	(1-32)
portID	(0-127)
drive-spec	*trayID*,*slotID* or *trayID*,*drawerID*,*slotID* A drive is defined as two or three integer literal values separated by a comma. Low-density trays require two values. High-density trays, those trays that have drawers, require three values.
drive-spec-list	*drive-specdrive-spec*
trayID-list	*trayID* {*trayID*}
esm-spec-list	*esm-spec* {*esm-spec*}
esm-spec	*trayID*, (left | right)
hex-literal	*0xhexadecimal-literal*
volumeGroup-number	*integer-literal*
filename	*string-literal*
error-action	(stop | continue)
drive-channel-identifier (four drive ports per tray)	(1 | 2 | 3 | 4)
drive-channel-identifier (eight drive ports per tray)	(1 | 2 | 3 | 4 | 5 | 6 | 7 | 8)
drive-channel-identifier-list	*drive-channel-identifier* {*drive-channel-identifier*}

Recurring Syntax	Syntax Value
host-channel-identifier (four host ports per tray)	(a1 | a2 | b1 | b2)
host-channel-identifier (eight host ports per tray)	(a1 | a2 | a3 | a4 | b1 | b2 | b3 | b4)
host-channel-identifier (16 host ports per tray)	(a1 | a2 | a3 | a4 | a5 | a6 | a7 | a8 | b1 | b2 | b3 | b4 | b5 | b6 | b7 | b8)
drive-type	(fibre | SATA | SAS) **NOTE** Only SAS drives are supported for firmware versions 7.86 and later.
drive-media-type	(HDD | SSD | unknown| allMedia) *HDD* means hard disk drive. *SSD* means solid state disk.
feature-identifier	(storagePartition2 | storagePartition4 | storagePartition8 | storagePartition16 | storagePartition64 | storagePartition96 | storagePartition128 | storagePartition256 | storagePartitionMax | snapshot | snapshot2 | snapshot4 | snapshot8 | snapshot16 | volumeCopy | goldKey | mixedDriveTypes | highPerformanceTier | SSDSupport | safeStoreSecurity | safeStoreExternalKeyMgr | dataAssurance) To use the High Performance Tier premium feature, you must configure a storage array as one of these: ■ SHIPPED_ENABLED ■ SHIPPED_ENABLED=FALSE; KEY_ENABLED=TRUE
repository-spec	*instance-based-repository-spec*|*count-based-repository-spec*

Recurring Syntax	Syntax Value
instance-based- repository-spec	(repositoryRAIDLevel =*repository-raid-level* repositoryDrives= (*drive-spec-list*) [repositoryVolumeGroupUserLabel =*user-label*] [trayLossProtect=(TRUE | FALSE)[1]]) | [drawerLossProtect=(TRUE | FALSE)[2]]) | (repositoryVolumeGroup=*user-label* [freeCapacityArea=*integer-literal*[3]]) Specify the repositoryRAIDLevel parameter with the repositoryDrives parameter. Do not specify the RAID level or the drives with the volume group. Do not set a value for the trayLossProtect parameter when you specify a volume group.
count-based-repository- spec	repositoryRAIDLevel =*repository-raid-level* repositoryDriveCount=*integer-literal* [repositoryVolumeGroupUserLabel =*user-label*] [driveType=*drive-type*[4]] [trayLossProtect=(TRUE | FALSE)[1]] | [drawerLossProtect=(TRUE | FALSE)[2]] | [dataAssurance=(none | enabled)[5]] |
wwID	*string-literal*
gid	*string-literal*
host-type	*string-literal* | *integer-literal*
host-card-identifier	(1 | 2 | 3 | 4)
backup-device-identifier	(1 | n | all) n is a specific slot number. Specifying **all** includes all of the cache backup devices availble to the entire storage array.
nvsram-offset	*hex-literal*
nvsram-byte-setting	*nvsram-value* = *0xhexadecimal* | *integer-literal* The *0xhexadecimal* value is typically a value from 0x0000 to 0xFFFF.
nvsram-bit-setting	*nvsram-mask, nvsram-value* = *0xhexadecimal, 0xhexadecimal* | *integer-literal* The *0xhexadecimal* value is typically a value from 0x0000 to 0xFFFF.
ip-address	(0-255).(0-255).(0-255).(0-255)
ipv6-address	(0-FFFF):(0-FFFF):(0-FFFF):(0-FFFF): (0-FFFF): (0-FFFF):(0-FFFF):(0-FFFF) You must enter all 32 hexadecimal characters.
autoconfigure-vols-attr-value-list	*autoconfigure-vols-attr-value-pair* {*autoconfigure-vols-attr-value-pair*}

Recurring Syntax	Syntax Value
autoconfigure-vols-attr-value-pair	driveType=*drive-type* | driveMediaType=*drive-media-type* | raidLevel=*raid-level* | volumeGroupWidth=*integer-literal* | volumeGroupCount=*integer-literal* | volumesPerGroupCount=*integer-literal*[6] | hotSpareCount=*integer-literal* | segmentSize=*segment-size-spec* | cacheReadPrefetch=(TRUE | FALSE) securityType=(none | capable | enabled)[7] | dataAssurance=(none | enabled)[5]
create-volume-copy-attr-value-list	*create-volume-copy-attr-value-pair* {*create-volume-copy-attr-value-pair*}
create-volume-copy-attr-value-pair	copyPriority=(highest | high | medium | low | lowest) | targetReadOnlyEnabled=(TRUE | FALSE) | copyType=(offline | online) | repositoryPercentOfBase=(20 | 40 | 60 | 120 | default) | repositoryGroupPreference=(sameAsSource | otherThanSource | default)
recover-raid-volume-attr-value-list	*recover-raid-volume-attr-value-pair* {*recover-raid-volume-attr-value-pair*}
recover-raid-volume-attr-value-pair	owner=(a | b) | cacheReadPrefetch=(TRUE | FALSE) | dataAssurance=(none | enabled)
cache-flush-modifier- setting	immediate, 0, .25, .5, .75, 1, 1.5, 2, 5, 10, 20, 60, 120, 300, 1200, 3600, infinite
serial-number	string-literal
usage-hint-spec	usageHint=(multiMedia | database | fileSystem)
iscsiSession	[*session-identifier*]
iscsi-host-port	(1 | 2 | 3 | 4) The host port number might be 2, 3, or 4 depending on the type of controller you are using.

Recurring Syntax	Syntax Value
ethernet-port-options	enableIPv4=(TRUE | FALSE) | enableIPv6=(TRUE | FALSE) | IPv6LocalAddress=*ipv6-address* | IPv6RoutableAddress=*ipv6-address* | IPv6RouterAddress=*ipv6-address* | IPv4Address=*ip-address* | IPv4ConfigurationMethod= (static | dhcp) | IPv4GatewayIP=*ip-address* | IPv4SubnetMask=*ip-address* | duplexMode=(TRUE | FALSE) | portSpeed=(autoNegotiate | 10 | 100 | 1000)
iscsi-host-port-options	IPv4Address=*ip-address* | IPv6LocalAddress=*ipv6-address* | IPv6RoutableAddress=*ipv6-address* | IPv6RouterAddress=*ipv6-address* | enableIPv4=(TRUE | FALSE) | enableIPv6=(TRUE | FALSE) | enableIPv4Priority=(TRUE | FALSE) | enableIPv6Priority=(TRUE | FALSE) | IPv4ConfigurationMethod= (static | dhcp) | IPv6ConfigurationMethod= (static | auto) | IPv4GatewayIP=*ip-address* | IPv6HopLimit=*integer* | IPv6NdDetectDuplicateAddress=*integer* | IPv6NdReachableTime=*time-interval* | IPv6NdRetransmitTime=*time-interval* | IPv6NdTimeOut=*time-interval* | IPv4Priority=*integer* | IPv6Priority=*integer* | IPv4SubnetMask=*ip-address* | IPv4VlanId=*integer* | IPv6VlanId=*integer* | maxFramePayload=*integer* | tcpListeningPort=*tcp-port-id* | portSpeed=(autoNegotiate | 1 | 10)
test-devices-list	*test-devices*{*test-devices*}
test-devices	controller=(a | b) esms=(*esm-spec-list*) drives=(*drive-spec-list*)
snapshot (legacy)-schedule-attribute-value-list	*snapshot (legacy)-schedule-attribute-value-pair*{*snapshot (legacy)-schedule-attribute-value-pair*}
time-zone-spec	(GMT+HH:MM | GMT-HH:MM) [dayLightSaving=HH:MM]

Recurring Syntax	Syntax Value
snapshot (legacy)-schedule-attribute-value-pair	startDate=MM:DD:YY scheduleDay=(*dayOfWeek* | all) startTime=HH:MM scheduleInterval=*interger* endDate=(MM:DD:YY | noEndDate) timesPerDay=*interger*

[1]For tray loss protection to work, the drives that comprise a volume group or a disk pool must be in separate trays. For RAID 3 or RAID 5 volume groups, no more than one drive must be in each tray. For RAID 6 volume groups, no more than two drives must be in each tray. For a disk pool, no more than two drives must be in each tray. When the `trayLossProtect` parameter is set to **TRUE** the storage array returns an error under these conditions:

- You have selected more than one drive from a tray in a RAID 3 or RAID 5 configuration.
- You have selected more than two drives from a tray in a RAID 6 configuration.
- You have selected more than two drives from a tray in a disk pool.

When the storage array is automatically selecting drives for a volume group or a disk pool, if the `trayLossProtect` parameter is set to **TRUE**, the storage array returns an error if the controller firmware cannot find drives that will enable the new volume group or the new disk pool to have tray loss protection. If the `trayLossProtect` parameter is set to **FALSE**, the storage array performs the operation even if it means that the volume group or disk pool might not have tray loss protection.

[2]In trays that have drawers for holding the drives, drawer loss protection determines whether data on a volume is accessible or inaccessible if a drawer fails. To help make sure that your data is accessible, set the `drawerLossProtect` parameter to **TRUE**. For RAID 3 or RAID 5 volume groups, no more than one drive must be in each drawer. For RAID 6 volume groups, no more than two drives must be in each drawer. For a disk pool, no more than two drives must be in each drawer.

If you have a storage array configuration in which a volume group spans several trays, you must make sure that the setting for drawer loss protection works with the setting for tray loss protection. You can have drawer loss protection without tray loss protection. You cannot have tray loss protection without drawer loss protection. If the `trayLossProtect` parameter and the `drawerLossProtect` parameter are not set to the same value, the storage array returns an error message and a storage array configuration will not be created.

[3]To determine if a free capacity area exists, run the **`show volumeGroup`** command.

[4]The default drive (drive type) is **`fibre`** (Fibre Channel).

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

[5]The `dataAssurance` parameter applies to the drives in a volume group. Using the `dataAssurance` parameter, you can specify that protected drives must be selected for a volume group. If you want to set the `dataAssurance` parameter to **`enabled`**, all of the drives in the volume group must be capable of data assurance. You cannot have a mix of drives that are capable of data assurance and drives that are not capable of data assurance in the volume group.

[6]The `volumesPerGroupCount` parameter is the number of equal-capacity volumes per volume group.

[7]The `securityType` parameter enables you to specify the security setting for a volume group that you are creating. All of the volumes are also set to the security setting that you choose. Available options for setting the security setting include:

- **none** – The volume group is not secure.
- **`capable`** – The volume group is security capable, but security has not been enabled.
- **`enabled`** – The volume group is security enabled.

NOTE A storage array security key must already be created for the storage array if you want to set `securityType=enabled`. (To create a storage array security key, use the **`create storageArray securityKey`** command.)

Usage Guidelines

This list provides guidelines for writing script commands on the command line:

- 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 parameters and secondary parameters with a space.
- The script engine is not case sensitive. You can enter commands by using uppercase letters, lowercase letters, or mixed-case letters.
- Add comments to your scripts to make it easier for you and future users to understand the purpose of the script commands. (For information about how to add comments, see Adding Comments to a Script File.)

NOTE While the CLI commands and the script commands are not case sensitive, user labels (such as for volumes, hosts, or host ports) are case sensitive. If you try to map to an object that is identified by a user label, you must enter the user label exactly as it is defined, or the CLI commands and the script commands will fail.

Adding Comments to a Script File

The script engine looks for certain characters or a command to show comments. You can add comments to a script file in three ways:

- Add text after two forward slashes (`//`) as a comment 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 appears, and the script operation is terminated. This error usually 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;
  ```

- Add text between **`/*`** and **`*/`** as a comment. If the script engine does not find both a starting comment notation and an ending comment notation, an error message appears, and the script operation is terminated.

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

- Use the **show** statement to embed comments in a script file that you want to appear while the script file is running. Enclose the text that you want to appear by using double quotation marks (“ ”).

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

Chapter 3 - Configuration Concepts

When you configure a storage array, you organize drives 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 chapter provides definitions of the physical and logical components required to organize the drives into a storage array configuration. This chapter also describes how the components relate to each other.

Before you begin to configure your storage array, become familiar with these concepts:

- Controllers
- Drives – including Full Disk Encryption-capable drives and drive security
- Hot spares
- Volume groups
- Disk pools
- Volumes
- RAID technology
- Hosts
- Host groups
- Host bus adapter (HBA) host ports
- Logical unit numbers (LUNs)

Configuring a RAID storage array requires caution and planning to make sure that you define the correct RAID level and the configuration for your storage array. The main purpose in configuring a storage array is to create volumes, which are addressable by the hosts, from a collection of drives. The commands described in the following chapters enable you to set up and run a RAID storage array. Additional commands also are available to provide you with more control and flexibility in managing and maintaining your storage array.

Controllers

All storage arrays have one or two controllers. The controllers are circuit-board assemblies that manage data flow and communications between the hosts and the drives in the storage array, keeping track of the logical addresses of where the data resides. 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 controller tray or a base system. A controller tray or a base system has two positions for controllers: slot A and slot B. The script commands identify each controller by the slot in which the controller is installed. If a controller tray or a base system has only one controller, the controller must be in slot A. A controller tray or a base system with two controllers is called a duplex tray. A controller tray or a base system with one controller is called a simplex tray.

Controllers manage the interface by running controller firmware to transmit and receive commands between the hosts and the drives. Host bus adapters facilitate the communication through whichever interface is selected. Typically, two host bus adapters and two paths are used to optimize redundancy.

A base system or a controller tray incorporates all host connections and expansion unit connections into each controller. The iSCSI host ports must be identified in your command statements to let you complete their network configurations.

Each controller in the E5460, E5424, E5412 controller-drive trays has four Fibre Channel host ports. Optionally, you can add a host interface card (HIC) for two InfiniBand (IB) host ports. Each controller has one SAS drive port.

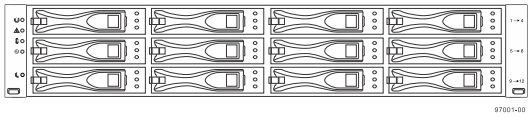
Each controller in the E2660, E2624, E2612 controller-drive trays can have two or four Fibre Channel host ports; two or four iSCSI host ports; or two SAS host ports. Each controller has one SAS drive port.

Drives

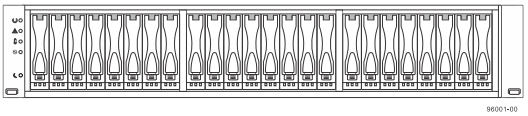
The drives are mounted in either a base system or an expansion unit. The base system has drives and one or, usually, two controllers in one tray. A expansion unit has drives, and is connected to a controller through an environmental services module (ESM). In addition to the drives and controllers or ESMs, the trays contain power supplies and fans. These components support base system and expansion unit operation and can be maintained through the CLI.

The base systems and expansion units can have one of these configurations:

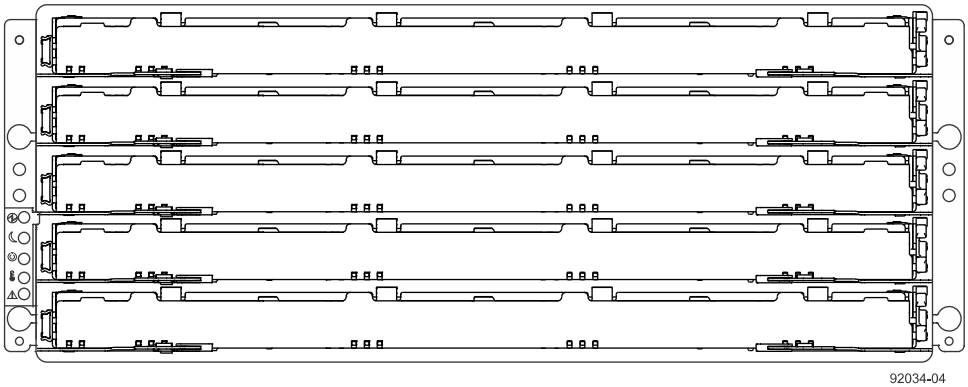
- 12 HDDs in a tray



- 24 HDDs in a tray

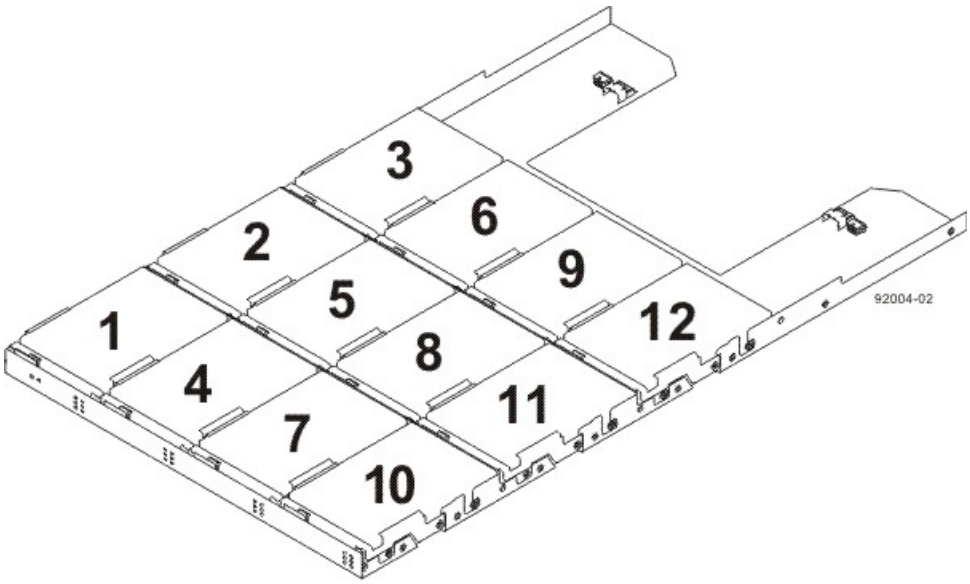


- 60 HDDs in a tray that has five drawers



In the 12-drive trays and the 24-drive trays the drives are located by tray ID and slot ID. In the 60-drive trays the drives are located by tray ID, drawer ID, and slot ID. The tray ID is the position of the tray in the storage array. Tray ID values are **0** to **99**. For older trays, you must set the tray ID values during installation using switches on the rear of the trays. For newer trays, the tray ID values are set automatically when power is applied to the storage array. The slot ID is the drive position in the drive tray. In drive trays with fewer than 60 drives, slot ID values range from **1** to **24**. In drive trays with 60 drives, slot ID values are defined by the drawer number and the position of the drive in the drawer. The drawer numbers range from **1** to **5**, counting from top to bottom. The position of each drive in a drawer is shown in the following figure.

Figure 1. Drive Drawer with Drives



The total number of drives in a storage array depends on the model of the base system and the capacity of the drives. The following table lists the maximum number of drives in a storage array, by base system model and expansion unit capacity.

Table 5. Maximum Number of Drives Supported

Base System with Expansion Units	Max # Drives
QS1200	96
QS2400	192
QD6000	480
QD7000	480

The maximum capacity in a storage array depends on the number of the drives in the storage array and the capacity of each drive in the storage array. The following table lists the maximum storage for each controller model based on the capacity of the drives.

Table 6. Maximum Capacity with Supported Drives

Drive Capacity	QS1200/QS2400/QD6000	QD7000
146-GB SAS	28.0 TB	
150-GB SAS	28.8 TB	
300-GB SAS	57.6 TB	
450-GB FC		201.0 TB
450-GB SAS	86.4 TB	
500-GB SAS	96.0 TB	
600-GB SAS	115.2 TB	
1.0-TB SATA		480.0 TB
1.0-TB SAS	192.0 TB	
2.0-TB SAS	384.0 TB	960.0 TB
3.0-TB SAS	576.0 TB	1440.0 TB
2.0-TB SATA		960.0 TB
3.0-TB SATA		1440.0 TB

Hot Spare Drives

NOTE Hot spare drives work only on storage arrays configured to use volume groups. Disk pools do not have hot spare drives.

A hot spare is a drive that acts as a standby in the event that a drive containing data fails. The hot spare is a drive 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 level 1, RAID level 3, RAID level 5, or RAID level 6. If a drive in a volume group fails, the controllers automatically replace the failed drive with a hot spare. The controllers use redundancy data to reconstruct the data from the failed drive onto the hot spare. To be most effective, the drive that you assign as a hot spare must have a capacity equal to or greater than the capacity of the largest drive in the storage array. Hot spares must meet the following conditions:

- The hot spare must be the same type of drive as the drive that failed. For example, a Serial Advanced Technology Attachment (SATA) hot spare cannot replace a Fibre Channel hot spare.
- The hot spare must have the same drive architecture as the drive that failed.

You can assign drives to act as hot spares manually or have the script commands automatically assign hot spares. If you manually assign a drive to be a hot spare, you must identify the drive by tray ID and slot ID. When you let the

script commands automatically assign hot spares, you must enter the number of hot spares that you want in the storage array.

Drive Security with Full Disk Encryption

Drive Security is a premium feature that prevents unauthorized access to the data on a drive that is physically removed from the storage array. Controllers in the storage array have a *security key*. Secure drives provide access to data only through a controller that has the correct security key. Drive Security is a premium feature of the storage management software and must be enabled either by you or your storage vendor.

NOTE Drive Security is not supported on all drives. To determine whether a drive supports Drive Security, select the **Hardware** tab in the SANtricity Storage Manager Array Management Window for your storage array and select the drive. In the **Security capable** field, you see either **Yes, Full Disk Encryption (FDE)** or **No**.

The Drive Security premium feature requires security capable drives. A security capable drive encrypts data during writes and decrypts data during reads. Each security capable drive has a unique drive encryption key.

When you create a secure volume group from security capable drives, the drives in that volume group become security enabled. When a security capable drive has been security enabled, the drive requires the correct security key from a controller to read or write the data. All of the drives and controllers in a storage array share the same security key. The shared security key provides read access and write access to the drives, while the drive encryption key on each drive is used to encrypt the data. A security capable drive works like any other drive until it is security enabled.

Whenever the power is turned off and turned on again, all of the security-enabled drives change to a *security locked* state. In this state, the data is inaccessible until the correct security key is provided by a controller.

You can view the Drive Security status of any drive in the storage array from the **Drive Properties** dialog. The drive can have one of these capabilities:

- Security Capable
- Secure – Security enabled or disabled
- Read/Write Accessible – Security locked or unlocked

You can view the Drive Security status of any volume group in the storage array by using the **`show volumeGroup`** command. The volume group can have one of these capabilities:

- Security Capable
- Secure

The following table shows how to interpret the security properties status of a volume group.

Table 7. Volume Group Security Properties

Security Status	Security Capable – yes	Security Capable – no
Secure – yes	The volume group is composed of all full disk encryption (FDE) drives and is in a Secure state.	Not applicable. Only FDE drives can be in a Secure state.
Secure – no	The volume group is composed of all FDE drives and is in a Non-Secure state.	The volume group is not entirely composed of FDE drives.

You can erase security-enabled drives so that you can reuse the drives in another volume group or in another storage array. Use the **`start secureErase`** command to completely erase any data on a security-enabled drive. Using the **`start secureErase`** command results in the loss of all of the data on a drive, and is irreversible. You can never recover the data.

The storage array password protects a storage array from potentially destructive operations by unauthorized users. The storage array password is independent from the Drive Security premium feature and should not be confused with the pass phrase that is used to protect copies of a security key. However, it is good practice to set a storage array password before you create, change, or save a security key or unlock secure drives.

Commands for FDE Drives

You can use these commands to enable security in the Full Disk Encryption (FDE) drives and manage the drives.

- **`create volume`** – Automatic drive select
- **`create volume`** – Free extent based select
- **`create volume`** – Manual drive select
- **`create storageArray securityKey`**
- **`create volumeGroup`**
- **`enable volumeGroup security`**
- **`export storageArray securityKey`**
- **`import storageArray securityKey`**
- **`set controller`**
- **`set storageArray securityKey`**
- **`show drive`**
- **`start secureErase`**

Volume Groups

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

NOTE Some storage arrays permit different drive types in the same tray; however, you cannot have a combination of different drives in the same volume group.

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

Capacity is the size of the volume group. Capacity is determined by the number of drives that you assign to the volume group. You can use only unassigned drives to create a volume group. (Storage space on unassigned drives 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 RAID level is the level of data protection that you want to define for your storage array. The RAID level that you choose affects storage capacity. When you configure your storage array, you must consider this compromise between

data protection and storage capacity. In general, the more protection that you need, the less storage capacity is available in your storage array.

The following table lists the minimum number of drives and the maximum number of drives that you can use in a volume group based on the RAID level that you want to assign to the volume group.

Table 8. Maximum Number of Drives in a Volume Group Based on RAID Level

RAID Level	Minimum Number of Drives	Maximum Number of Drives	Redundancy
0	1	All, if drive count is even; All - 1, if drive count is odd.	None
1, 10	2	All, if drive count is even; All - 1, if drive count is odd.	Mirrored pairs
3	3	30	1 drive
5	3	30	1 drive
6	5	30	2 drives

You can determine the size of the volume group by multiplying the maximum number of drives in the volume group by the capacity of the smallest drive in the volume group.

Disk Pools

A disk pool is a set of drives that is logically grouped together in the storage array. The drives in each disk pool must be of the same drive type and drive media type, and they must be similar in size. As with a volume group, you can create one or more volumes in the disk pool. However, the disk pool is different from the volume group in the way that the data is distributed across the drives that comprise the disk pool.

In a volume group, the data is distributed across the drives based on a RAID level. You can specify the RAID level when you create the volume group. The data for each volume is written sequentially across the set of drives in the volume group.

In a disk pool, the storage management software distributes the data for each volume randomly across a set of drives in the disk pool. Each disk pool must have a minimum of eleven drives. Although there is no limit on the maximum number of drives that can comprise a disk pool, the disk pool cannot contain more drives than the maximum limit for each storage array. The storage management software automatically configures the RAID level when you create the disk pool. You cannot set or change the RAID level of disk pools or the volumes in the disk pools.

NOTE Because disk pools can co-exist with volume groups, a storage array can contain both disk pools and volume groups.

Disk Pool Benefits

- **Better use of drives** – When you add drives to a storage array, the storage management software automatically detects the drives and prompts you to create a single disk pool or multiple disk pools based on the drive type and the current configuration. If disk pools were previously defined, the storage management software provides the option of adding the compatible drives to an existing disk pool. When new drives are added to an existing disk pool, the storage management software automatically redistributes the data across the new capacity, which now

includes the new drives that you added. The data in the volumes remain accessible when you add the drives to the disk pool. When you delete disk pool volumes, the capacity of those volumes is added to the total usable capacity of the disk pool and, therefore, can be reused.

NOTE You have the option to manually create a disk pool, if you prefer not to proceed with the automatic disk pool creation process.

- **Reduced hot spots** – A host might access some drives in the volume group for data more frequently than other drives because of the sequential manner in which the data is written to the drives. This frequency of access to drives creates hot spots in the volume group. In a disk pool, the hot spots are significantly reduced because of the random manner in which the data is spread across a large number of drives. The reduction of hot spots in the disk pool improves performance of the storage array.
- **Faster reconstruction of data** – Disk pools do not use hot spare drives for data protection like a volume group does. Instead of hot spare drives, disk pools use spare capacity within each drive that comprises the disk pool.

  In hot spare drive coverage, the maximum drive IOPS limits the speed of reconstruction of data from the failed drive to the hot spare drive. In a disk pool, the reconstruction of data is much faster because the spare capacity in all of the drives that comprise the disk pool is used. Additionally, the data to reconstruct after a drive failure is reduced because the data is spread randomly across more drives in a disk pool.

  Faster reconstruction of data in a disk pool also reduces the risk of additional drive failures during a reconstruction operation. For example, consider a drive failure in a RAID level 5 volume group that is comprised of three drives. The time it takes to reconstruct the data from the failed drive is relatively longer for a volume group. During the reconstruction of data, if another drive fails in this volume group, data loss occurs. Unlike volume groups, the time for which the disk pool is exposed to multiple drive failures during a reconstruction operation is significantly reduced.
- **Reduced maintenance** – You can configure the storage management software to send alert notifications when the configured capacity of a disk pool is reaching a specified percentage of free capacity. Additionally, you do not need to manage any hot spare drives. You can replace a set of drives during a scheduled maintenance of the storage array.

Disk Pool Restrictions

- You cannot change the RAID level of a disk pool. The storage management software automatically configures disk pools as RAID level 6.
- You cannot export a disk pool from a storage array or import the disk pool to a different storage array.
- All drive media types in a disk pool must be the same.
- You can protect your disk pool with Full Disk Encryption (FDE), but the drive attributes must match. For example, FDE-enabled drives cannot be mixed with FDE-capable drives. You can mix FDE-capable and non FDE-capable drives, but the encryption abilities of the FDE drives cannot be used.
- If you downgrade the controller firmware version of a storage array that is configured with a disk pool to a firmware version that does not support disk pools, the volumes are lost and the drives are treated as unaffiliated with a disk pool.

Volumes

A volume is the logical component that hosts use for data storage. Hosts that are attached to the storage array write data to the volumes and read data from the volumes. You can create a volume from either a volume group or a disk pool. Before you create a volume, the volume group or a disk pool must already exist and it must have enough free capacity to create the volume.

Table 9. Volume Properties Set Using CLI Commands

Properties of a Standard Volume	Description
Segment size	The amount of data stored on a drive before the storage array moves to the next drive in the stripe (RAID group). Applies only to volumes in volume groups, not disk pools.
Capacity	The amount of data that you can store in a volume.
Controller ownership	Defines the controller that is designated to be the owning, or primary, controller of the volume. Controller ownership is very important and should be planned carefully. Make sure that the controllers are balanced as closely as possible for total I/Os.
Mapping	How host LUNs are mapped to a volume.
Name	Descriptive name indicating the type of data stored in the volume.

A volume is a contiguous subsection of a volume group or disk pool 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 that users choose. The volume names can be any combination of alphanumeric characters, hyphens (-), and underscores (_). The maximum length of a volume name is 30 characters.

The script commands support the following types of volumes:

- **Standard volume** – 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.
- **Thin volume** – A logical structure in which the volumes have small physical storage allocations, but a large virtual capacity available for host I/O data writes. When you configure a thin volume, you specify two types of capacity: the virtual capacity and the repository capacity. The virtual capacity is the capacity that is reported to the host. The repository capacity is the amount of physical drive space that is currently allocated for writing data.
- **Access volume** – A factory-configured volume in a storage area network (SAN) environment that is used for communication between the storage management software and the storage array controller. The access volume uses a logical unit number (LUN) address and consumes 20 MB of storage space. The 20 MB of access volume storage space is not available for data storage.

NOTE Use the access volume only for in-band-managed storage arrays.

- **Snapshot (Legacy) volume** – A logical point-in-time image of another volume. A snapshot (legacy) 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 copies those blocks to a snapshot (legacy) repository volume.
- **Snapshot (Legacy) repository volume** – A special volume in the storage array that is created as a resource for a snapshot (legacy) volume. A snapshot (legacy) repository volume contains snapshot (legacy) data and copy-on-write data for a particular snapshot (legacy) volume.
- **Base volume** – A standard volume from which you create a snapshot (legacy) 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 (legacy) volume.
- **Primary volume** – A standard volume in a Synchronous Mirroring 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 Synchronous Mirroring 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 a 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 Synchronous Mirroring configuration that is created as a resource for each controller in both the local storage array and the 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 shutdown of the storage arrays.

The number and capacity of the volumes in your storage array depends on the type of controller in the storage array. The following table lists the maximum number of volumes in a storage array that each controller model supports.

Table 10. Maximum Number of Volumes Each Controller Model Supports

Specification	QS1200/QS2400	QD6000/QD7000
Maximum number of volumes per storage array	2048	2048
Maximum number of volumes per volume group	256	256
Maximum number of volumes per disk pool	2048	2048
Maximum volume size for a volume group	Number of drives supported by array x (capacity of smallest supported drive by array – 512 MB)	
Maximum volume size for a disk pool	64 TB	64 TB
Maximum number of drives per volume group using RAID level 5	30	30
Maximum number of drives per disk pool using RAID level 6	192	192
Maximum number of remote mirrors	16	16

NOTE The maximum volume size is limited by the size of the drives and the configuration of the storage array. The last 512 MB on each drive is reserved for storage array configuration database and potential future expansion. For practical considerations, you want to constrain the maximum volume size so that drive replacement and volume reconstruction does not take an excessive amount of time.

RAID Levels

The RAID level defines a storage architecture in which the storage capacity on the drives in a volume group is separated into two parts: part of the capacity stores the user data, and the remainder stores redundant or parity information about the user data. The RAID level that you choose determines how user data is written to and retrieved from the drives. You can define five RAID levels: RAID level 0, RAID level 1, RAID level 3, RAID level 5, and RAID level 6. Each level provides different performance and protection features.

RAID Level 0 provides the fastest storage access but does not provide any redundant information about the stored data. RAID level 1, RAID level 3, RAID level 5, and RAID level 6 write redundancy information to the drives to provide fault tolerance. The redundancy information might be a copy of the data or an error-correcting code that is derived from the data. In RAID level 1, RAID level 3, RAID level 5, or RAID level 6 configurations, if a drive fails, the redundancy information can be used to reconstruct the lost data. Regardless of the RAID level that 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. The following table lists the RAID levels and describes the configuration capabilities of each level.

Disk pools are automatically configured to RAID level 6 by the storage management software. You cannot change the RAID level for disk pools.

Table 11. RAID Level Configurations

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

RAID Level	Configuration
6	**Data protection or continuous access mode** – RAID level 6 stripes both user data and redundancy data (in the form of parity) across the drives. A minimum of five drives are required for a RAID level 6 volume group. The equivalent capacity of two drives is used for the redundancy data. Two different algorithms calculate redundancy data, which are in the form of both a P parity and a Q parity. RAID level 6 works well for larger drive sizes. Recovery from a second drive failure in the same volume group is possible. If two drives fail in a RAID level 6 volume group, all of the associated volumes become degraded, but the redundancy data permits the data to be reconstructed. If three or more drives fail, all of the associated volumes fail, and all data is lost.

Hosts

A host is a computer that is attached to the storage array for accessing the volumes in the storage array. The host is attached to the storage array through HBA host ports, which are connectors on 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 that users choose. The host name can be any combination of alphanumeric characters, hyphens, and underscores. The maximum length of the host name is 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 lets the controllers in the storage array adapt their behavior (such as LUN reporting and error conditions) to the operating system of the host that is sending the information. Host types are identified by a label or an index number that is generated by the controller 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 that users choose. The host group name can be any combination of alphanumeric characters with a maximum length of 30 characters.

Host Bus Adapter Host Ports

A host bus adapter (HBA) provides the physical connection from the host to the storage array. The host port is a physical connector on an HBA. The HBA is a circuit board that is installed in the host. The HBA can have one or more host ports. Each host port is identified by a unique, 16-byte World Wide Identifier (WWID). If the HBA has more than one host port, each host port has a unique ID.

When you first turn on the power to a storage array, the storage management software automatically detects the HBA host ports. Initially, all detected host ports belong to a default group. You can use script commands to identify the WWIDs on a storage array and, if you choose, change them. If you move an HBA 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 HBA host ports that you can logically define for your storage array depends on the type of controller in the storage array. The following table lists the maximum number of HBA host ports that you can define.

Table 12. Maximum Number of HBA Host Ports per Controller

Base System	Max # SAS Host Ports	Max # IB Host Ports
QS1200	128	256
QS2400	128	256
QD6000	128	256
QD7000	128	256

Table 13. HBA Speed

Base System	Base Port	Optional HIC
QS1200	8 Gb FC	• 6 Gb SAS • 8 Gb FC • 10 Gb iSCSI • 40 Gb IB
QS2400	8 Gb FC	• 6 Gb SAS • 8 Gb FC • 10 Gb iSCSI • 40 Gb IB
QD6000	8 Gb FC	• 6 Gb SAS • 8 Gb FC • 10 Gb iSCSI • 40 Gb IB
QD7000	None	• 10 Gb iSCSI • 12 Gb SAS • 16 Gb FC • 56 Gb IB

Logical Unit Numbers

A logical unit number (LUN) is a unique value that identifies the volumes in a storage array. The hosts identify the volumes that they want to access using the LUN values. When you create a volume, the firmware assigns the LUN values, 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 that you specify must be available on every host in the host group.

Chapter 4 - Configuring a Storage Array

When you configure a storage array, you organize drives 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. You want to maximize the data availability by making sure that 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 many items, including but not limited to the following conditions:

- The redundant array of independent disks (RAID) level for the volume group
- The settings for the segment size and the cache block size
- Whether the cache read prefetch capability is turned on or turned off

Data protection is determined by the RAID level, hardware redundancy (such as global hot spares for volume group), and software redundancy (such as the Synchronous Mirroring premium feature and the snapshot premium features).

The sections in this chapter show some, but not all, of the CLI wrapper commands and the script commands. The commands in this chapter show how you can use the commands to configure a storage array. These presentations do not describe all possible usage and syntax for the commands. For complete definitions of the commands, including syntax, parameters, and usage notes, see Chapter 1 - About the Command Line Interface.

NOTE Many of these commands require a thorough understanding of the firmware as well as an understanding of the network components that need to be mapped. Use the CLI commands and the script commands with caution.

This chapter contains examples of CLI command usage and script command usage. The command syntax that is used in the examples is for a host running a Windows operating system. As part of the examples, the complete `C:\` prompt and the path for the commands are shown. Depending on your operating system, the prompt and path construct can vary.

For most commands, the syntax is the same for all UNIX operating systems and Windows operating systems, as well as for a script file. UNIX operating systems, however, may apply non-literal meanings to certain characters used on the command line. On UNIX operating systems, you use a backslash (`\`) immediately before that character to indicate that the next character is a literal character. For example, to use the name "Engineering" with the quote marks as string literal characters, precede the two quote marks with a backslash, as in `\"Engineering\"`.

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 features and software features that are on the storage array. When you configure a storage array that has an existing configuration, you must make sure 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 use unassigned drives to create a new volume group or a new disk pool. Before you create a new volume group or a new disk pool, you must determine which drives are available. The commands that are described in this section help you to determine the components and the 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.

To return the most information about the storage array, run the **`show storageArray`** command with the `profile` parameter. This example shows the complete CLI command and script command:

```
c:\...\smX\client>smcli 123.45.67.88
-c “show storageArray profile;”
```

This example identifies the storage array by the IP address `123.45.67.88` . This address is the IP address of one of the controllers in the storage array. You also can identify the storage array by name.

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

```
c:\...\smX\client>smcli 123.45.67.88
-c “show storageArray profile;” -o
c:\folder\storagearrayprofile.txt
```

In this example, the name `folder` is the folder in which you choose to place the profile file, and `storagearrayprofile.txt` is the name of the file. You can choose any folder and any file name.

ATTENTION Possible loss of data – When you write information to a file, the script engine does not check to determine if the file name already exists. If you choose the name of a file that already exists, the script engine writes over the information in the file without warning.

To return a brief list of the storage array features and components, use the `summary` parameter. The command looks like this example:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “show storageArray summary;”
```

The following example shows the type of information that is returned by the **`show storageArray`** command with the `summary` parameter.

```
PROFILE FOR STORAGE ARRAY: jlane_SM_ie (Mon Sep 22 15:09:38 CDT 2014)
STORAGE ARRAY-----------------------------
STORAGE ARRAY INFORMATION AND SETTINGS
Storage array world-wide identifier (ID): 60080E50002471C20000000054099F28
Chassis Serial Number:                    1144FG000121
Event configuration data version:         112#NTAP_1120_8
CACHE SETTINGS
Start demand cache flushing at:  80%
Cache block size:                4 KB
Media scan frequency:               Disabled
Failover alert delay:               5 minutes
AUTOSUPPORT SUMMARY
Status:              Enabled
Schedule information
Daily schedule:    Not Available
Weekly schedule:   Not Available
.
.
.
```

The **show** commands return information about the specific components of a storage array. The information returned by each of the **show** commands is the same as the information returned by the **`show storageArray profile`** command, but the information is constrained to the specific component. Some **show** commands are designed to run on a specific device or controller. For a complete list of **show** commands, see *Command Line Interface and Script Commands Programming Guide*.

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. This command deletes all of the existing configuration information, including all of the volume groups, disk pools, volumes, host mappings, and hot spare definitions from the controller memory. Use the **`clear storageArray configuration`** command only when you create a new configuration.

ATTENTION Possible damage to the storage array configuration – As soon as you run this command, the existing storage array configuration is deleted. Before running this command, make sure that you save support data.

The command has this form:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “clear storageArray configuration;”
```

This command has two parameters that you can use to limit the amount of configuration information removed:

- `all` – Removes the entire configuration of the storage array, including security information and identification information. Removing all of the configuration information returns the storage array to its initial state.
- `volumeGroups` – Removes the storage array mapping (volume configuration, volume group configuration, disk pools, and thin volumes), but leaves the rest of the configuration intact.

If you want to create new volume groups and volumes within the storage array but do not want to clear the rest of the configuration, you should use the **`clear storageArray configuration`** command with the `volumeGroups` parameter to remove existing volume groups in a pre-existing configuration.

Configuring a Storage Array with Volume Groups

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

NOTE Many of these commands require a thorough understanding of the firmware as well as an understanding of the network components that need to be mapped. Use the CLI commands and the script commands with caution.

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 these parameters:

- The media type of drives (HDD)
- The RAID level
- The number of drives in a volume group
- The number of volume groups
- The number of volumes in each volume group

- The number of hot spares
- The size of each segment on the drives
- A cache read prefetch

After you define these parameters, the SANtricity Storage Manager software creates the volume groups, the volumes, and the hot spares. The controllers assign volume group numbers and volume numbers as they are created. After the SANtricity Storage Manager 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 autoConfiguration`** command. The **`show storageArray autoConfiguration`** command returns a list of parameter values that the SANtricity Storage Manager software uses 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 that the **`show storageArray autoConfiguration`** command returns, run the **`autoConfigure storageArray`** command without new parameter values.

The **`autoConfigure storageArray`** command has this form:

```
autoConfigure storageArray
driveType=(fibre | SATA | SAS)
raidLevel=(0 | 1 | 3 | 5 | 6)
volumeGroupWidth=numberOfDrives
volumeGroupCount=numberOfVolumeGroups
volumesPerGroupCount=numberOfVolumesPerGroup
hotSpareCount=numberOfHotSpares
segmentSize=segmentSizeValue
cacheReadPrefetch=(TRUE | FALSE)
securityType=(none | capable | enabled)
dataAssurance=(none | enabled)]
```

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

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

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

The `hotSpareCount` parameter defines the number of hot spares that you want in the array.

The `segmentSize` parameter defines the amount of data, in KB, that the controller writes on a single drive in a volume before writing data on the next drive. The smallest units of storage are data blocks. A data block stores 512 bytes of data. The size of a segment determines how many data blocks that it contains. An 8-KB segment holds 16 data blocks. A 64-KB segment holds 128 data blocks.

Using a single drive for a single request leaves other drives available to simultaneously service other requests. Valid segment size values are **`8`**, **`16`**, **`32`**, **`64`**, **`128`**, **`256`**, and **`512`**.

NOTE If you set the cache block size to **`16`**, you cannot create a volume with a segment size of **`8`**.

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 drives in the volume group that are used for data storage. In this environment, multiple drives are used for the same request, but each drive is accessed only once.

The `cacheReadPrefetch` parameter turns on or turns off the ability of the controller to read additional data blocks into the cache. When you turn on cache read prefetch, the controller copies additional data blocks into the cache while it is reading requested data blocks from a drive into the cache. This action increases the chance that a future request for data can be fulfilled from the cache, which improves the speed with which data is accessed. Cache read prefetch is important for applications that use sequential I/O, such as multimedia applications.

Valid values for the `cacheReadPrefetch` parameter are **TRUE** or **FALSE**. If you want to turn on cache read prefetch, set the `cacheReadPrefetch` parameter to **TRUE**. If you want to turn off cache read prefetch, set the `cacheReadPrefetch` parameter to **FALSE**.

The following table lists the default values for the segment size and cache read prefetch settings for different storage array uses.

Table 14. Default Values for Segment Size and Cache Read Prefetch

Storage Array Use	Segment Size (KB)	Cache Read Prefetch
File system	**128**	**TRUE**
Database	**128**	**TRUE**
Multimedia	**256**	**TRUE**

Use the `securityType` parameter when you have security-capable drives that can support the Drive Security premium feature (FDE). This parameter enables you to specify the security level when you create the volume group that uses the security-capable drives. The settings for the `securityType` parameter are the following:

- **none** – The volume group and volumes are not secure.
- **`capable`** – The volume group and volumes are capable of having security set, but security has not been enabled.
- **`enabled`** – The volume group and volumes are security enabled.

After you have finished creating the volume groups and the volumes by using the **`autoConfigure storageArray`** command, you can further define the properties of the volumes in a configuration by using the **`set volume`** command.

Example of the Auto Configuration Command

```
c:\...\smX\client>smcli 123.45.67.88
-c “autoConfigure storageArray driveMediaType=hdd
raidLevel=5 volumeGroupWidth=8 volumeGroupCount=3
volumesPerGroupCount=4 hotSpareCount=2
segmentSize=8 cacheReadPrefetch=TRUE;”
```

The command in this example creates a storage array configuration by using hard disk drives set to RAID level 5. Three volume groups are created, and each volume group consists of eight drives, which are configured into four volumes. The storage array has two hot spares. The segment size for each volume is 8 KB. The cache read prefetch is turned on, which causes additional data blocks to be written into the cache.

Using the Create Volume Command

Use the **`create volume`** command 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 drives.

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

You must have unconfigured capacity in the volume group. You do not need to assign the entire capacity of the volume group to a volume.

Creating Volumes with User-Assigned Drives

When you create a new volume and assign the drives 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 command has this form:

```
create volume drives=((trayID1, slotID1... trayIDn, slotIDn)|
(trayID,drawerID,slotID... trayID,drawerID,slotID))
raidLevel=(0 | 1 | 3 | 5 | 6) userLabel="volumeName"
volumeGroupUserLabel=["volumeGroupName"]
[capacity=volumeCapacity owner=(a | b)
cacheReadPrefetch=(TRUE | FALSE)
segmentSize=segmentSizeValue]
```

NOTE The `capacity` parameter, the `owner` parameter, the `cacheReadPrefetch` parameter, and the `segmentSize` parameter are optional parameters (indicated by the placement inside the square brackets). You can use one or all of the optional parameters as needed to define your configuration. If you choose not to use any of the optional parameters, the default values of the parameters are used for your configuration. See *Command Line Interface and Script Commands Programming Guide* for a list of default values.

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

The `drives` parameter is a list of the drives that you want to use for the volume group.

For DE6600 drive trays	For DE5600 and DE1600 drive trays
Enter the tray ID, drawer ID, and slot ID of each drive that you want to use. Enclose the list in parentheses, separate the tray ID value, the drawer ID value, and the slot ID value of a drive with commas. Separate each tray ID, drive ID, and slot ID group with a space. This example shows you how to enter tray ID values and slot ID values: `(1,1,1 1,1,2 1,1,3 1,1,4 1,1,5)`	Enter the tray ID and the slot ID of each drive that you want to use. Enclose the list in parentheses, separate the tray ID value and the slot ID value of a drive with a comma. Separate each tray ID and slot ID pair with a space. This example shows you how to enter tray ID values and slot ID values: `(1,1 1,2 1,3 1,4 1,5)`

The `capacity` parameter defines the size of the volume. You do not need to assign the entire capacity of the drives to the volume. Later, you can assign any unused space to another volume.

NOTE If the `capacity` parameter is not specified, the entire available capacity of the volume group is used by default.

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

The `cacheReadPrefetch` parameter and the `segmentSize` parameter are the same as those described for the **`autoConfigure storageArray`** command.

Example of Creating Volumes with User-Assigned Drives

```
c:\...\smX\client>smcli 123.45.67.88
-c “create volume drives=(1,1 1,2 1,3 2,1 2,2 2,3)
raidLevel=0 userLabel=\”Engineering_1\” capacity=512GB
owner=a cacheReadPrefetch=TRUE segmentSize=128;”
```

The command in this example automatically creates a new volume group and a volume with the name Engineering_1. The volume group uses RAID level 0. The command uses six drives to construct the volume group. The capacity of the volume will be 3 TB, which is distributed across all six drives. If each drive has a capacity of 4 TB, the total capacity of all the assigned disks is 24 TB.

```
4 TB x 6 drives = 24 TB
```

Because only 3 TB is assigned to the volume, 21 TB remains available (as unconfigured capacity) for other volumes that a user can add to this volume group later.

```
24 TB - 3 TB volume group size = 21 TB
```

NOTE Depending on which RAID level you use, some space might need to be allocated for parity, so the unconfigured capacity might be reduced more than this example illustrates.

Cache read prefetch is turned on, which causes additional data blocks immediately above the host-requested read range to be placed in cache. The segment size for this volume group is 28 KB. Tray loss protection is set to **TRUE**, which means that if a tray fails, there are enough remaining drives in the volume group to satisfy host I/O requests.

Creating Volumes with Software-Assigned Drives

If you choose to let the storage management software assign the drives when you create the volume, you need only to specify the number of drives that you want to use. The storage management software then assigns the drives. The controller firmware assigns a volume group number to the new volume group. To manually create volume groups and volumes, use the **`create volume`** command:

```
create volume driveCount=numberOfDrives
raidLevel=(0 | 1 | 3 | 5 | 6)
userLabel=volumeName
[driveMediaType=(hdd | ssd | allMedia | unknown)]
[capacity=volumeCapacity | owner=(a | b) |
cacheReadPrefetch=(TRUE | FALSE) |
segmentSize=segmentSizeValue]
[trayLossProtect=(TRUE | FALSE)]
```

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

Example of Creating Volumes with Software-Assigned Drives

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “create volume driveCount=6 raidLevel=5
userLabel=\”Engineering_1\”
capacity=20GB owner=a cacheReadPrefetch=TRUE segmentSize=128;”
```

The command in this example creates the same volume as the example for the previous **`create volume`** command in which a user assigns the drives. The difference is that a user does not know which drives 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 this command:

```
create volume volumeGroup=volumeGroupNumber
userLabel=volumeName
[freeCapacityArea=freeCapacityIndexNumber |
capacity=volumeCapacity | owner=(a | b) |
cacheReadPrefetch=(TRUE | FALSE) |
segmentSize=segmentSizeValue]
```

NOTE Parameters wrapped in square brackets or curly brackets are optional. You can use one or all of the optional parameters as needed to define your configuration. If you choose not to use any of the optional parameters, the default values of the parameter are provided for your configuration.

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 get 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, hyphens, and underscores. The maximum length of the volume name is 30 characters. You must enclose the volume name with double quotation marks (“ ”).

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 free capacity area to use for volume creation. You do not have to assign the entire capacity of the drives to the volume. Later, you can assign any unused space to another volume.

The use of the `capacity` parameter, the `owner` parameter, the `cacheReadPrefetch` parameter, and the `segmentSize` parameter is the same as described in the previous examples of the **`create volume`** command.

Tray Loss Protection

NOTE Tray loss protection works only with volume groups.

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 drive in a volume group must be on a separate tray. The way in which tray loss protection works depends on the method that you choose to assign the drives for a volume group.

When you assign the drives, if you set `trayLossProtect`=**TRUE** and have selected more than one drive from any one tray, the storage array returns an error. If you set `trayLossProtect`=**FALSE** or the parameter is absent, the

drives are selected without regard for tray loss protection, and that selection may or may not result in a set of drives with tray loss protection.

When the controller firmware assigns the drives, if `trayLossProtect=`**TRUE**, the storage array posts an error if the controller firmware cannot provide drives that result in the new volume group having tray loss protection. If `trayLossProtect=`**FALSE**, the storage array performs the operation even if it means that the volume group might not have tray loss protection.

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

Configuring a Storage Array with Disk Pools

A disk pool is a collection of 11 or more drives in a storage array that have the same spindle speed, the same security level, and preferably the same capacity to make the most efficient use of the drives.

A storage array can contain one or more disk pools, although the benefits of using a disk pool increase as the number of drives in a disk pool increase. Creating a disk pool with the largest number of similar drives is the preferred approach. However, if not all drives in the storage array have the same characteristics or if you want certain drives to support different applications, you can create more than one disk pool on your storage array. There is no practical limit on the number of drives that can comprise a disk pool, although a disk pool cannot contain more drives than the maximum limit for each storage array.

Two differences between setting up a disk pool and a volume group are that with a disk pool the RAID level is preset to RAID 6 and there is no need to designate a hot spare drive. In a disk failure condition in a disk pool, instead of using hot spare technology all the operational drives in the disk pool participate in the reconstruction process. The reconstruction data space is used to reconstruct the data from the failed drive. With a volume group, hot spare technology is used to recover from a drive failure condition and you select the RAID level during the configuration process.

In general, you configure a storage array by defining a disk pool, the volumes, and which hosts have access to the volumes. The following sections explain how to use the script commands to perform the general steps to create a disk pool configuration from an array of drives.

You can have two types of volumes in a disk pool:

- Standard volume

  A standard volume has a fixed capacity that you can define when you create the volume.
- Thin volume

  A thin volume is an expandable volume with both physical capacity and virtual capacity.

NOTE Many of these commands require a thorough understanding of the firmware as well as an understanding of the network components that need to be mapped. Use the CLI commands and the script commands with caution.

Using the Create Disk Pool Command

Use the **`create diskPool`** command to create a new disk pool in two ways:

- Create a new disk pool manually by selecting and assigning drives to the disk pool.

You must have unassigned drives in the storage array. You do not need to assign the entire capacity of the storage array to the disk pool. You can mix both disk pools and volume groups in a storage system.

When you create a new disk pool you want to assign the largest number of drives possible to the disk pool. The more drives that you have in a disk pool, the more robust the disk pool is, with faster rebuild times and simplified management requirements. The minimum number of drives that you can have in a disk pool is 11; the maximum number of drives is as many as required to support the maximum number of volumes that a controller can manage. The following table shows the drive counts and the number of drives reserved for reconstruction.

Table 15. Number of Drives in a Disk Pool to Support RAID 6

Drive Count	Drives Reserved for Reconstruction	Comments
11	1	Minimum number of drives in a disk pool
12-31	2	
32-63	3	
64-127	4	Minimum number of drives in a QS1200 system is 96
128-191	5	
192-255	6	Minimum number of drives in a QS2400 system is 192
256-384	7	
385-512	8	Minimum number of drives in a QD6000/ QD7000 system is 480

The only supported RAID level for a disk pool is RAID 6, with a stripe width of 10. The storage array must have a minimum drive count to create a pool. The number of drives in the pool also influences how much reserved capacity is needed to redistribute data for rebuilds.

Hot spares are not required or allowed for disk pools. Spare capacity for reconstruction is divided among the drives within a disk pool. A small amount of each drive is reserved as reconstruction space to hold reconstructed data in the event of loss of access to a drive or a drive failure. Because of this behavior, the system can sustain drive failures until the capacity is exhausted or the number of drives in the disk pool falls below the minimum drive count. As long as free capacity exists on the system, failed drives are rebuilt and degraded volumes are brought back to optimal. The only constraint to rebuilding failed drives is the RAID level of the volumes in the disk pool.

In volume groups, the RAID level is determined when the volume group is created. Each volume defined on the volume group inherits the same RAID level. The RAID stripe width in a volume group is determined by the number of drives in the volume group. Because of the way that the disk pool volume data is mapped onto the disk pool, the stripe width is independent of the number of drives in the pool.

In your storage array, you want to configure as few disk pools as are required so that every drive in the storage array is included as a member of one of the disk pools. Reasons for having several disk pools in a storage array might include separate disk pools to use certain types of drives, or to create different pools for different applications. For the most efficient disk pool, all of the drives need to have the same characteristics:

- Spindle speed
- Capacity

With both the Security feature and the Protection Type feature, drives are either capable or non-capable. You can mix security-capable and non-security-capable drives in the same disk pool, but none of the drives in the pool will be able to use the security feature if there are any non-security-capable drives in the disk pool. Similarly, you can mix PI-capable and non-PI-capable drives in the same disk pool, but protection information is not available unless all drives in the disk pool are PI-capable.

Before you create a disk pool, run the **`show storageArray`** command to determine the drives that are available and to make sure that you have enough drives in the storage array to create a disk pool.

The **`create diskPool`** commands have several optional parameters to enable you to create the configuration that you want. The use of these parameters is shown in the examples in the following sections. For more information about the use of the optional parameters refer to the *Command Line Interface and Script Commands* guide.

After you create the disk pool, you can create the volumes.

Creating Disk Pools with Software-Assigned Drives

If you choose to let the storage management software assign the drives when you create a disk pool, you need only to specify a name (user label) for the disk pool. Optionally, you also can specify the number of drives that you want to use. The storage management software then assigns the drives. The controller firmware assigns a number to the new disk pool. To create a disk pool with software-assigned drives use this command:

```
create diskPool driveType=(fibre|sas)
userLabel="diskPoolName"
[driveCount=driveCountValue |
warningThreshold=(warningThresholdValue|default) |
criticalThreshold=(criticalThresholdValue|default) |
criticalPriority=(highest|high|medium|low|lowest) |
backgroundPriority=(highest|high|medium|low|lowest) |
degradedPriority=(highest|high|medium|low|lowest) |
securityType=(none|capable|enabled) |
driveMediaType=(hdd | ssd | allMedia | unknown) |
dataAssurance=(none|enabled)]
```

Example of Creating Volumes with Software-Assigned Drives

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “create diskpool driveType=sas userLabel=”Engineering_1”
driveCount=64 warningthreshold=65 criticalthreshold=75
criticalpriority=high backgroundpriority=medium
degradedpriority=high securitytype=enabled drivemediatype=hdd
dataassurance=enabled;”
```

This command creates a disk pool with these features:

- Type of drive is SAS.
- The name of the disk pool is Engineering_1. The disk pool name can be any combination of alphanumeric characters, hyphens, and underscores. The maximum length of the disk pool name is 30 characters. You must enclose the disk pool name with double quotation marks (“ ”).
- The storage management software adds 64 drives to the disk pool. This assumes that the storage array has a minimum of 64 drives that have the same characteristics.
- When the disk pool consumes 65 percent of its capacity, a warning alert is posted. The default value is 50 percent. The `warningthreshold` parameter must always be set to a value lower than the `criticalthreshold` parameter.
- When the disk pool consumes 75 percent of its capacity, a critical alert is posted. The default value is 85 percent.

- The priority for reconstruction operations for critical events on the disk pool is set to high. If a critical condition occurs, such as two drives failing at the same time, the storage management software makes the reconstruction of the data a high priority.
- The priority for background operations on this disk pool is set to medium. If a background operation occurs, such as reconstruction or formatting, the background operation equally shares resources with other storage array operations.
- The priority for correcting the disk pool after it has entered a Degraded state is set to high. If a condition occurs, such as a single drive failure, the storage management software makes the correction of the condition a high priority.
- The `securitytype` is enabled, so the storage management software uses only drives that are configured to be security drives.
- The type of drive to be used is a hard drive (hdd).
- The disk pool uses only drives with protected data capability.

Creating Disk Pools with User-Assigned Drives

In some situations you might be required to create a disk pool by assigning the drives to the disk pool instead of having the software assign the drives. One situation might be when you want to create a small disk pool to test possible configuration parameters. Another situation might be when you do not have enough drives with the same characteristics to create a disk pool. If all of the usable drives in the storage array do not have the same capacity you can still configure a disk pool by manually selecting the drives. One constraint, however, is that each drive in the disk pool assumes the same capacity as the smallest drive, even if the other drives have more capacity.

NOTE You can mix drive capacities, and you can mix security-capable and non-security-capable drives in the same disk pool. You cannot mix spindle speeds, and you cannot mix HDD and SSD drives in the same disk pool.

The command to manually assign drives has this form:

```
create diskPool drives=(trayID1,drawerID1,slotID1 ... trayIDN,drawerIDN,slotIDN
userLabel="diskPoolName"
[driveCount=driveCountValue |
warningThreshold=(warningThresholdValue|default) |
criticalThreshold=(criticalThresholdValue|default) |
criticalPriority=(highest|high|medium|low|lowest) |
backgroundPriority=(highest|high|medium|low|lowest) |
degradedPriority=(highest|high|medium|low|lowest) |
securityType=(none|capable|enabled) |
driveMediaType=(hdd | ssd | allMedia | unknown) |
dataAssurance=(none|enabled)]
```

The `drives` parameter is a list of the drives that you want to use for the disk pool. Enter the tray ID and the slot ID of each drive that you want to use. For high-capacity drive trays that have drawers to hold the drives, also use the drawer number. For high-capacity drive trays the sequence of the location identifiers is drive tray, drawer, slot. Enclose the list in parentheses, separate the tray ID value, drawer ID value, and the slot ID value of a drive with a comma, and separate each tray ID, drawer ID, and slot ID set with a space. This example shows you how to enter tray ID values and slot ID values for low capacity drive trays:

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

This example shows you how to enter tray ID values, drawer ID values, and slot ID values for high capacity drive trays:

```
(1,1,1 1,2,3 1,3,5 1,4,6 1,5,8)
```

Example of Creating Volumes with User-Assigned Drives

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “create diskpool drives=(1,1,1 1,1,2 1,2,3 ... 2,1,10, 2,2,11)
userLabel=”Engineering_1” warningthreshold=65 criticalthreshold=75
criticalpriority=high backgroundpriority=medium
degradedpriority=high securitytype=enabled drivemediatype=hdd
dataassurance=enabled;”
```

This command creates a disk pool with these features:

- The list of drives represents the drives found in a high capacity drive tray.
- The name of the disk pool is Engineering_1. The disk pool name can be any combination of alphanumeric characters, hyphens, and underscores. The maximum length of the disk pool name is 30 characters. You must enclose the disk pool name with double quotation marks (“ ”).
- When you assign drives to a disk pool, you do not need to use the `driveCount` parameter.
- When the disk pool consumes 65 percent of its capacity, a warning alert is posted. The default value is 50 percent. The `warningthreshold` parameter always must be set to a value lower than the `criticalthreshold` parameter.
- When the disk pool consumes 75 percent of its capacity, a critical alert is posted. The default value is 85 percent.
- The priority for reconstruction operations for critical events on the disk pool is set to high. If a critical condition occurs, such as two drives failing at the same time, the storage management software makes the reconstruction of the data a high priority.
- The priority for background operations on this disk pool is set to medium. If a background operation occurs, such as reconstruction or formatting, the background operation equally shares resources with other storage array operations.
- The priority for correcting the disk pool after it has entered a Degraded state is set to high. If a condition occurs, such as a single drive failure, the storage management software makes the correction of the condition a high priority.
- The `securitytype` is enabled, so the storage management software uses only drives that are configured to be security drives.
- The type of drive to be used is a hard drive (hdd).
- The disk pool uses only drives with protected data capability.

Using the Create Volume Command

The **`create volume diskPool`** command enables you to create a volume in a disk pool. You can create either one of these types of volumes:

- Standard volume
- Thin volume

A standard volume has a fixed capacity that you can define when you create the volume. The standard volume reports only the fixed capacity to the host. In disk pools, the volume capacity is distributed across all of the applicable drives. You do not need to identify specific drives for the volume.

A thin volume is an expandable volume with both physical capacity and virtual capacity. Physical capacity is the size of the volume at a given time that is currently allocated for writing data. This size can increase over time. Virtual

capacity is capacity reported to the hosts and is the "size" of the volume. Thin provisioning enables you to create volumes with a large virtual capacity and relatively small physical capacity, which is beneficial for storage utilization and efficiency. Thin volumes can help simplify storage administration because the physical capacity can increase as the application needs change, without disrupting the application, allowing for better storage utilization.

Keep these guidelines in mind when choosing a name for your volume:

- A volume name can consist of letters, numbers, and the special characters underscore (_), hyphen (-), and pound (#). If you choose any other characters, an error message appears. You are prompted to choose another name.
- Limit the name to 30 characters. Any leading and trailing spaces in the name are deleted.
- Use a unique, meaningful name that is easy to understand and remember.
- Avoid arbitrary names or names that would quickly lose their meaning in the future.
- If you choose a volume name that duplicates that of another volume, an error message appears. You are prompted to choose another name.

Data assurance protection (DA) checks for and corrects errors that might occur as data is communicated between a host and a storage array. Enabling this feature for a new volume helps make sure that errors are detected and corrected. To enable DA, these conditions must be in place:

- The storage array must be able to support DA.
- The disk pool must be DA capable.

If any of the host connections on the controllers in your storage array do not support DA, the associated hosts cannot access data on DA-enabled volumes. Before creating a DA-enabled volume, make sure that the host connection that you are planning to use supports DA.

Dynamic cache read prefetch allows the controller to (optionally) copy additional sequential data blocks into the cache while it is reading data blocks from a drive to cache. This caching increases the chance that future requests for data can be filled from the cache. Cache read-ahead is important for multimedia applications that use sequential I/O. The rate and amount of data that is prefetched into cache is self-adjusting based on the rate and request size of the host reads. Random access does not cause data to be prefetched into cache. This feature has no effect when read caching is disabled.

NOTE When you are creating a thin volume, the dynamic cache read prefetch option is not available.

Creating Standard Volumes on a Disk Pool

If you want to add a new standard volume to an existing disk pool, use this command:

```
create volume diskPool="diskPoolName"
userLabel="volumeName"
capacity=volumeCapacity
thinProvisioned=FALSE |
[owner=(a|b) |
mapping=(none|default) |
dataAssurance=(none|enabled) |
cacheReadPrefetch=(TRUE | FALSE)]
```

NOTE The `thinProvisioned` parameter is optional. If not specified, the default value is FALSE.

The `diskPool` parameter is the name of the disk pool in which you want to create a new volume. If you do not know the disk pool names on the storage array, you can use the **`show storageArray summary`** command to get a list of the disk pool.

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

The `capacity` parameter defines the capacity to use for the volume.

The `thinProvisioned` parameter sets the volume to either standard or thin. For a standard volume, the `thinProvisioned` parameter must be set to **FALSE**, or the parameter can be omitted, which makes it default to **FALSE**.

The `owner` parameter defines which controller is designated to be the primary controller owner of the volume. For best performance make sure that the controllers are balanced as closely as possible for total I/Os.

The `mapping` parameter defines whether you want the storage management software to map the volume to a host, or if you want to map the volume to a host at a later time. To allow the storage management software to map the volume to a host, use the `default` parameter. To map the volume to a host later, use the `none` parameter. To map a volume to a host, use the **`set volume logicalUnitNumber`** command.

To turn on data assurance, set the `dataAssurance` parameter to **`enabled`**.

To turn on cache read prefetch, set the `cacheReadPrefetch` parameter to **`TRUE`**.

The usage of the `owner` parameter, the `cacheReadPrefetch` parameter, and the `segmentSize` parameter is the same as described in the previous examples of the **`create volume`** command.

Creating Thin Volumes on a Disk Pool

If you want to add a new thin volume to an existing disk pool, use this command:

```
create volume diskPool="diskPoolName"
userLabel="volumeName" capacity=volumeVirtualCapacity
thinProvisioned=TRUE
[owner=(a|b) |
mapping=(none|default) |
dataAssurance=(none|enabled) |
(existingRepositoryLabel=existingRepositoryName |
newRepositoryCapacity=newRepositoryCapacityValue [KB | MB | GB | TB | Bytes]) |
repositoryMaxCapacity=repositoryMaxCapacityValue[KB|MB|GB|TB|Bytes] |
repositoryExpansionPolicy=(automatic|manual) |
warningThresholdPercent=warningThresholdPercentValue |
cacheReadPrefetch=(TRUE | FALSE)]
```

NOTE The `thinProvisioned` parameter is required to create a thin volume on a disk pool.

The `diskPool` parameter is the name of the disk pool in which you want to create a new thin volume. If you do not know the disk pool names on the storage array, you can use the **`show allVolumes summary`** command to get a list of the volumes and the disk pools to which the volumes belong.

The `userLabel` parameter is the name that you want to give to the disk pool. The disk pool name can be any combination of alphanumeric characters, hyphens, pound (#), and underscores. The maximum length of the disk pool name is 30 characters. You must enclose the disk pool name with double quotation marks (“ ”).

The `capacity` parameter defines the virtual capacity of the thin volume. The capacity is the value that is reported to the host. As users add information to the thin volume the physical size of the volume increases. When you define the capacity of the thin volume you must define a capacity of at least 32 GB. The maximum capacity that you can define is 64 TB.

The `mapping` parameter defines whether you want the storage management software to map the volume to a host, or if you want to map the volume to a host later. To allow the storage management software to map the volume to a host use the `default` parameter. To map the volume to a host later, use the `none` parameter. To map a volume to a host, use the **`set volume logicalUnitNumber`** command.

The repository capacity is the actual physical capacity of the thin volume. The value that you use for the repository capacity is the starting size of the physical component of a thin volume. The minimum capacity that you can define for the repository must be at least 4 GB. The maximum capacity that you can define is 64 TB. You can use a small starting value for the repository. As data increases in the repository, additional standard volumes are added to the repository to increase the capacity. You can either use an existing repository or create a new repository with this command. The repository capacity is governed by these parameters:

- `existingRepositoryLabel` – Use this parameter when you want to associate the volume with an existing repository volume.
- `newRepositoryCapacity` – Use this parameter when you want to create a new repository volume.
- `repositoryMaxCapacity` – Use this parameter to define the maximum size that you want for the repository volume.
- `repositoryExpansionPolicy` – Use this parameter to define whether the repository volume expands automatically or if you need to expand the repository volume.

The `warningThresholdPercent` parameter defines when you receive a warning that the repository volume is nearing maximum capacity. The value for this parameter is a percentage of the maximum capacity of the repository volume.

To turn on cache read prefetch, set the `cacheReadPrefect` parameter to **`TRUE`**.

NOTE Thin volumes do not use data assurance.

Modifying Your Configuration

For most configurations, after you have created your initial configuration by using the **`autoConfigure storageArray`** command or the **`create volume`** command, you must modify the properties of your configuration to make sure that 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 these properties:

- The controller clocks
- The storage array password
- The storage array host type
- The storage array cache
- The global hot spares

Setting the Controller Clocks

To synchronize the clocks on the controllers with the management station, use the **`set storageArray time`** command. Run this command to make sure that event time stamps that are written by the controllers to the Event Log match the event time stamps that are written to the host log files. The controllers stay available during synchronization. This example shows the command:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “set storageArray time;”
```

Setting the Storage Array Password

Use the **`set storageArray`** command to define a password for a storage array. The command has this form:

```
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.

ATTENTION Possible data corruption or data loss – Implementing destructive commands can cause serious damage, including data loss.

Unless you define a password for the storage array, you can run all of the script commands. A password protects the storage array from any command that the controllers consider destructive. A destructive command is 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 characters. You must enclose the password in double quotation marks (“ ”). This example shows how to use the **`set storageArray`** command to define a password:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “set storageArray password=”1a2b3c4d5e”;”
```

Setting the Storage Array Host Type

Use the **`set storageArray`** command to define the default host type. The command has this form:

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

The `defaultHostType` parameter defines how the controllers in the storage array communicate with the operating system on undefined hosts that are connected to the storage array SAN. This parameter defines the host type only for data I/O activities of the storage array. This parameter does not define the host type for the management station.

For example, if you set the `defaultHostType` parameter to **`Linux`**, the controller communicates 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 that you might need to use this parameter is when you need to change how the storage array behaves relative to the hosts that are connected to it.

This example shows how to define a specific default host type:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “set storageArray defaultHostType=11;”
```

The value **11** is the host type index value from the host type table that appears after entering the previous command.

Setting the Storage Array Cache

The cache is high-speed memory that holds data that is either written to the drives or read by the host.

The cache acts as a buffer so that data transfers between the host and the drive do not need to be synchronized. In read caching, the data for a read operation from the host might already be in the cache from a previous operation, which eliminates the need to access the drives. The data stays in the read cache until it is overwritten by newer read or write cache data. For write caching, a write operation stores data from the host in cache until it can be written to the drives.

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

- **set storageArray**
- **set volume**

Use the **set storageArray** command to change the cache block size, the cache flush start value, and the cache stop value. The command has this form:

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

You can enter one, two, or all three of the parameters on the command line.

The cache block size value defines the size of the data block that is used by the controller in transferring data into or out of the cache. You can set the cache block size to either **4KB**, **8KB**, **16KB** or **32KB**. The value that you use applies to the entire storage array and all of the volumes in the storage array. For redundant controller configurations, this value includes all volumes owned by both controllers. Use smaller cache block sizes for systems that require transaction processing requests or I/O streams that are typically small and random. Use larger cache block sizes for large I/O, sequential, high-bandwidth applications. The choice of block size affects read/write performance. Large data transfers take longer in 4-KB block sizes than 16-KB block sizes. This example shows how to set the `cacheBlockSize` parameter:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “set storageArray cacheBlockSize=16;”
```

To prevent data loss or corruption, the controller periodically writes cache data to the drives (flushes the cache) when the amount of unwritten data in the cache reaches a predefined level, called a start percentage. The controller also writes cache data to the drives when data has been in the cache for a predetermined amount of time. The controller writes data to the drives until the amount of data in the cache drops to a stop percentage level. Use the **set storageArray** command to set the start value as a percentage 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. This example shows how to set these parameters:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “set storageArray cacheFlushStart=80;”
```

Low start percentages provide for maximum data protection. For low start percentages, the chance that data requested by a read command is not in the cache is increased. When the data is not in the cache, the cache hit

percentage for writes and I/O requests decreases. A low start value also increases the number of writes that are necessary to maintain the cache level. Increasing the number of writes increases the system overhead and further decreases performance.

Use the **`set volume`** command to change settings for the cache flush modifier, cache without batteries enabled, mirror cache enabled, the read ahead multiplier, read cache enabled, and write cache enabled. Use this command to set properties for all of the volumes or for a specific volume in a volume group. The command has this form:

```
set (allVolumes | volume [volumeName] |
volumes [volumeName1 ... volumeNameN]
volume <wwID>) |
cacheWithoutBatteryEnabled=(TRUE | FALSE) |
mirrorCacheEnabled=(TRUE | FALSE) |
readCacheEnabled=(TRUE | FALSE) |
writeCacheEnabled=(TRUE | FALSE) |
cacheReadPrefetch=(TRUE | FALSE)
```

The `cacheWithoutBatteryEnabled` parameter 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 uninterruptible power supply (UPS) and you enable this parameter, you can lose data if power to the storage array fails. This example shows how to set this parameter value:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “set volume [\”Engineering\”]
cacheWithoutBatteryEnabled=FALSE;”
```

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

NOTE The **`mirrorCacheEnabled`** parameter is ignored unless the **`writeCacheEnabled`** parameter is set to **TRUE**.

To use this option, these conditions must exist:

- 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**. This example shows how to set this parameter:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “set volume [\”Accounting\”] mirrorCacheEnabled=TRUE;”
```

The `readCacheEnabled` parameter 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 drives and places them in the cache. Until the cache is flushed, all of the other requests for this data are fulfilled with cache data rather than from a read, which increases throughput. To enable read caching, set this parameter to **TRUE**. To disable read caching, set this parameter to **FALSE**. This example shows how to set this parameter:

```
c:\...\sm9\client>smcli 123.45.67.88 123.45.67.89
-c “set volume [\”Balance_04\”] readCacheEnabled=TRUE;”
```

The `writeCacheEnabled` parameter changes the time when the host is notified that the write is complete. If the `writeCacheEnabled` parameter is set to **TRUE**, the host is notified that the write is complete as soon as the data has been written to cache. If the `writeCacheEnabled` parameter is set to **FALSE**, the host is notified that the write is complete when the data is written to non-cache media. To enable write caching, set this parameter to **TRUE**. To disable write caching, set this parameter to **FALSE**. This example shows how to set this parameter:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “set allVolumes writeCacheEnabled=TRUE;”
```

The `cacheReadPrefetch` parameter turns on or turns off the ability of the controller to read additional data blocks into cache. When you turn on cache read prefetch, the controller copies additional data blocks into cache while it is reading requested data blocks from a drive into cache. This action increases the chance that a future request for data can be fulfilled from the cache, which improves the speed with which data is accessed. Cache read prefetch is important for applications that use sequential I/O, such as multimedia applications.

Valid values for the `cacheReadPrefetch` parameter are **TRUE** or **FALSE**. If you want to turn on cache read prefetch, set the `cacheReadPrefetch` parameter to **TRUE**. If you want to turn off cache read prefetch, set the `cacheReadPrefetch` parameter to **FALSE**. This example shows how to set this parameter:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “set volume [\”Engineering_1\” \”Engineering_2\”]
cacheReadPrefetch=TRUE;”
```

Assigning Global Hot Spares

NOTE You can assign hot spares for volume groups. You cannot assign hot spares for disk pools.

You can assign or unassign global hot spares by using the **`set drive`** command. To use this command, you must identify the location of the drives by either the tray ID and slot ID for standard capacity trays, or by tray ID, drawer ID and slot ID for high capacity trays. Then, you set the `hotSpare` parameter to **TRUE** to enable the hot spare or **FALSE** to disable an existing hot spare. The command has this form for high capacity trays:

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

For standard capacity trays omit the `drawerID` value.

This example shows how to set hot spare drives:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “set drives [1,1,2 1,2,3] hotSpare=TRUE;”
```

Enter the tray ID, drawer ID, and slot ID of each drive that you want to use. Enclose the list in square brackets, separate the tray ID value, drawer ID value, and slot ID value of a drive with commas, and separate each tray ID, drawer ID, and slot ID triplet with 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 by using the **`save storageArray configuration`** command. Saving the configuration creates a script file that you can run on the command line. The command has this form:

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

ATTENTION Possible loss of data – When information is written to a file, the script engine does not check to determine if the file name already exists. If you choose the name of a file that already exists, the script engine writes over the information in the file without warning.

You can choose to save the entire configuration or specific configuration features. This example shows how to set this parameter value:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “save storagearray configuration
file=\”c:\\folder\\storagearrayconfig1.scr\”;”
```

In this example, the name `folder` is the folder in which you want to place the profile file and `storagearrayconfig1.scr` is the name of the file. You can choose any folder and any file name. The file extension for a configuration file is `.scr`. The storage management software uses this extension when it creates the configuration file.

Chapter 5 - Using the Snapshot (Legacy) Premium Feature

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

Snapshot (Legacy) volumes provide these capabilities:

- Create a complete image of the data on a base volume at a particular point in time
- Use only a small amount of storage space
- Provide for quick, frequent, non-disruptive backups, or testing new versions of a database system without affecting real data
- Provide for snapshot (legacy) volumes to be read, written, and copied
- Use the same availability characteristics of the base volume (such as RAID protection and redundant path failover)
- Map the snapshot (legacy) volume and make it accessible to any host on a storage area network (SAN). You can make snapshot (legacy) data available to secondary hosts for read access and write access by mapping the snapshot (legacy) to the hosts
- Create up to 16 snapshot (legacy)s per volume and up to 1024 snapshots per storage array. The maximum number of snapshots depends on the model of the controller. The maximum number of snapshot (legacy) volumes is one-half of the total number of volumes that are supported by the controller.
- Increase the capacity of a snapshot (legacy) volume
- Schedule snapshots (legacy) for:
  - Times when storage array usage is low
  - Times for regular recurring snapshot (legacy) creation

How Snapshot (Legacy) Works

Three components comprise a snapshot (legacy) volume: the base volume, the snapshot (legacy) volume, and the snapshot (legacy) repository volume. The following table lists the components and briefly describes what they do.

Table 16. Components of a Snapshot (Legacy) Volume

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

Based on information that you provide through the script commands, the storage management software creates an empty snapshot (legacy) repository volume and defines the mapping from a base volume to the snapshot (legacy)

repository volume. The snapshot (legacy) repository volume holds changed data that a host writes to the base volume. When the snapshot (legacy) repository volume is first created, it holds only the metadata about the snapshot (legacy) volume with which it is associated.

NOTE When you first create a snapshot (legacy) repository volume, briefly stop all of the 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 is also copied to the snapshot (legacy) repository volume. This action is called copy-on-write. A snapshot (legacy) is constructed by combining the updated data in the snapshot (legacy) repository volume with data in the base volume that has not been altered. This action creates a complete copy of the base volume at a specific point in time. The snapshot (legacy) appears as a volume that contains the original data at the time of creation, but the snapshot (legacy) is actually an image that is the combination of the snapshot (legacy) repository volume and the original base volume. The snapshot (legacy) repository volume, which houses original data that has been changed, is the only additional drive space that is needed for the snapshot (legacy) volume. The additional drive space is typically 10 percent to 20 percent of the drive space of the base volume and varies depending on the amount of changes to the data. The longer a snapshot (legacy) volume is active, the larger the snapshot (legacy) repository volume must be. The default size of the snapshot (legacy) repository volume is 20 percent of the base volume; however, you can set the size of the snapshot (legacy) repository volume to other values.

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

Table 17. Snapshot (Legacy) Volume Commands

Command	Description
`create snapshotVolume`	This command creates a snapshot (legacy) volume.
`recreate snapshot`	This command starts a fresh copy-on-write operation by using an existing snapshot (legacy) volume.
`recreate snapshot collection`	This command restarts multiple snapshot (legacy) volumes as one batch operation by using one or many existing snapshot (legacy) volumes.
`set (snapshotVolume)`	This command defines the properties for a snapshot (legacy) volume and lets you rename a snapshot (legacy) volume.
`stop snapshot`	This command stops a copy-on-write operation.

About Scheduling Snapshots (Legacy)

You can create a snapshot (legacy) volume to perform snapshot (legacy) operations at a later time or at regularly occurring intervals by adding a schedule to the snapshot (legacy) volume. If you do not add a schedule to the snapshot (legacy) volume, the snapshot (legacy) operation occurs immediately. You can add a schedule when you create a snapshot (legacy) volume or you can add a schedule to an existing snapshot (legacy) volume. Each snapshot (legacy) volume can have only one schedule.

You can create a schedule that runs daily or weekly in which you select specific days of the week (Sunday through Saturday).

You can also create a schedule that runs monthly in which you select specific days of the month.

Typical Uses

Scheduled backups – For example, an application stores business-critical data in two volumes on the storage array. You back up this data every work day at 11:00 p.m. To accomplish this type of backup, select the first volume. Create a schedule that runs one time per day on Monday, Tuesday, Wednesday, Thursday, and Friday. Choose a time between the end of your work day and 11:00 p.m. Select a starting date of today and no end date. Apply this schedule to the second volume also. Map the two snapshot (legacy) volumes to your backup host, and perform the regular backup procedures. Unmap the two snapshot (legacy) volumes before the next scheduled snapshot (legacy) operation time. If you do not unmap the snapshot (legacy) volumes, the storage array will skip the next snapshot (legacy) operation to avoid data corruption.

Rapid recovery – In this example, you back up your data at the end of every work day and keep hourly snapshots (legacy) from 8:00 a.m. to 5:00 p.m. If data loss or corruption occurs during the work day, you can recover the data from the snapshots (legacy) so that the data loss window is smaller than one hour. To accomplish this, create a schedule containing a start time of 8:00 a.m. and an end time of 5:00 p.m. Select 10 snapshots (legacy) per day on Monday, Tuesday, Wednesday, Thursday, and Friday. Select a start date of today and no end date. Create an end of day backup as described in the "Scheduled backups" example.

Guidelines

Keep the following guidelines in mind when creating schedules for snapshot (legacy) volumes:

- You can either create a schedule when you create a snapshot (legacy) volume, or you can add a schedule to an existing snapshot (legacy) volume.
- Scheduled snapshot (legacy) operations do not occur when the snapshot (legacy) volume is mapped, the storage array is offline or powered off, or the snapshot (legacy) volume is used as a source volume in a Volume Copy operation and the status of the copy operation is Pending or In progress.
- If you delete a snapshot (legacy) volume that has a schedule, the schedule is also deleted.
- Schedules are stored in the configuration database on the storage array.

Snapshot (Legacy) Scheduling Commands

The following table lists the commands for setting and managing schedules.

Table 18. Snapshots (Legacy) Scheduling Commands

Command	Description
`create snapshotVolume`	This command creates a snapshot (legacy) volume. Using this command, you can define how you want to schedule snapshots (legacy).
`set (snapshotVolume)`	This command defines the properties for a snapshot (legacy) volume and lets you reset the parameters for a snapshot (legacy) schedule.
`start snapshot rollback`	This command starts a rollback operation, which immediately changes the contents of a base volume to match a specific point-in-time version of the base volume.

Command	Description
stop snapshot rollback	This command stops a rollback operation. **Attention** – Stopping a snapshot (legacy) rollback can leave the base volume and snapshot (legacy) volume unusable.
resume snapshot rollback	This command resumes a rollback operation that was paused by the action of the controller.

Creating a Snapshot (Legacy) Volume

The **create snapshotVolume** command provides three methods for defining the drives for your snapshot (legacy) repository volume:

- Defining the drives for the snapshot (legacy) repository volume by their tray IDs and their slot IDs.
- Defining a volume group in which the snapshot (legacy) repository volume resides. In addition, you can define the capacity of the snapshot (legacy) repository volume.
- Defining the number of drives, but not specific drives, for the snapshot (legacy) repository volume.

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

Creating a Snapshot (Legacy) Volume with User-Assigned Drives

Creating a snapshot (legacy) volume by assigning the drives provides flexibility in defining your configuration by letting you choose from the available drives in your storage array. When you choose the drives for your snapshot (legacy) volume, you automatically create a new volume group. You can specify which drives to use and the RAID level for the new volume group. The command has this form:

```
create snapshotVolume baseVolume="baseVolumeName"
(repositoryRAIDLevel=(1 | 3 | 5 | 6)
repositoryDrives=(trayID1,drawerID1,slotID1 ... trayIDn,drawerIDn,slotIDn))
[repositoryVolumeGroupUserLabel="repositoryVolumeGroupName"
trayLossProtect=(TRUE | FALSE)
drawerLossProtect=(TRUE | FALSE)
freeCapacityArea=freeCapacityIndexNumber
userLabel="snapshotVolumeName"
warningThresholdPercent=percentValue
repositoryPercentOfBase=percentValue
repositoryUserLabel="repositoryName"
repositoryFullPolicy=(failBaseWrites | failSnapshot) |
enableSchedule=(TRUE | FALSE) |
schedule=(immediate | snapshotSchedule)]
```

This example shows a command in which users assign the drives:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
```

```
-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 (legacy) of the base volume Mars_Spirit_4. The snapshot (legacy) repository volume consists of five drives that form a new volume group. The new volume group has RAID level 5. This command also takes a snapshot (legacy) of the base volume, which starts the copy-on-write operation.

This example shows how to use the command in a script file:

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

This example shows a minimal version of the command:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “create snapshotVolume baseVolume=\”Mars_Spirit_4\”;”
```

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

This example shows how to use the command in a script file:

```
create snapshotVolume baseVolume=“Mars_Spirit_4”;
```

Creating a Snapshot (Legacy) Volume with Software-Assigned Drives

With this version of the **`create snapshotVolume`** command, you choose an existing volume group in which to place the snapshot (legacy) repository volume. The storage management software determines which drives to use. You can also define how much space to assign to the snapshot (legacy) repository volume. Because you are using an existing volume group, the RAID level for the snapshot (legacy) volume defaults to the RAID level of the volume group in which you place it. You cannot define the RAID level for the snapshot (legacy) volume. The command has this form:

```
create snapshotVolume baseVolume="baseVolumeName"
[repositoryVolumeGroup="repositoryVolumeGroupName"
repositoryUserLabel="repositoryName"
freeCapacityArea=freeCapacityIndexNumber
userLabel="snapshotVolumeName"
warningThresholdPercent=percentValue
repositoryPercentOfBase=percentValue
repositoryFullPolicy=(failBaseWrites | failSnapshot) |
trayLossProtect=(TRUE | FALSE)
enableSchedule=(TRUE | FALSE) |
schedule=(immediate | snapshotSchedule)]
```

This example shows a command in which the storage management software assigns the drives:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “create snapshotVolume baseVolume=\”Mars_Spirit_4\”
repositoryVolumeGroup=2 freeCapacityArea=2;”
```

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

When you define the capacity of a snapshot (legacy) repository volume, specify a size that is 20 percent of the size of the base volume. In the previous example, the size of the snapshot (legacy) repository volume is set to 4 GB. The underlying assumption is that the base volume size is 20 GB (0.2 x 20 GB= 4 GB).

This example shows how to use the command in a script file:

```
create snapshotVolume baseVolume=”Mars_Spirit_4”
repositoryVolumeGroup=2 freeCapacityArea=2;
```

Creating a Snapshot (Legacy) Volume by Specifying a Number of Drives

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

This example shows how to use a command in which users specify the number of drives:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “create snapshotVolume baseVolume=\”Mars_Spirit_4\”
repositoryRAIDLevel=5 repositoryDriveCount=3;”
```

The command in this example creates a new snapshot (legacy) repository volume that consists of three drives. Three drives comprise a new volume group that has RAID level 5. This command also takes a snapshot (legacy) of the base volume, which starts the copy-on-write operation.

This example shows how to use the command in a script file:

```
create snapshotVolume baseVolume= “Mars_Spirit_4”
repositoryRAIDLevel=5 repositoryDriveCount=3;
```

User-Defined Parameters

Use the parameters in the **`create snapshotVolume`** command to define the snapshot (legacy) volume to suit the requirements of your storage array. The following table lists the parameters and briefly describes what the parameters do.

Table 19. Snapshot (Legacy) Volume Parameters

Parameter	Description
`driveType`	The type of drive that you want to use for the snapshot (legacy) repository volume. The choice is **`fibre`** (Fibre Channel), **SATA**, or **SAS**. This parameter works only with the count-based repository method of defining a snapshot (legacy) volume.

Parameter	Description
`repositoryVolumeGroup`	The volume group in which you want to build the snapshot (legacy) repository volume. The default value is to build the snapshot (legacy) repository volume in the same volume group as the base volume.
`freeCapacityArea`	The amount of storage space that you want to use for the snapshot (legacy) repository volume. Free storage space is defined in units of bytes, KB, MB, GB, or TB.
`userLabel`	The name that you want to give to the snapshot (legacy) volume. If you do not choose a name for the snapshot (legacy) volume, the software creates a default name by using the base volume name. For example, with a base volume name of Mars_Spirit_4: ■ When the base volume does not have a snapshot (legacy) volume, the default snapshot (legacy) volume name is Mars_Spirit_4-1. ■ When the base volume already has *n*-1 number of snapshot (legacy) volumes, the default name is Mars_Spirit_4-*n*.
`repositoryUserLabel`	The name that you want to give to the snapshot (legacy) repository volume. If you do not choose a name for the snapshot (legacy) repository volume, the software creates a default name by using the base volume name. For example, if the base volume name is Mars_Spirit_4 and does not have an associated snapshot (legacy) repository volume, the default snapshot (legacy) repository volume name is Mars_Spirit_4-R1. If the base volume already has *n*-1 number of snapshot (legacy) repository volumes, the default name is Mars_Spirit_4-R*n*.
`warningThresholdPercent`	The percentage of the capacity that you will permit the snapshot (legacy) repository volume to get before you receive a warning that the snapshot (legacy) repository volume is nearing full. The warning value is a percentage of the total capacity of the snapshot (legacy) repository volume. The default value is **`50`**, which represents 50 percent of the total capacity. (You can change this value later by using the **`set snapshotVolume`** command.)
`repositoryPercentOfBase`	The size of the snapshot (legacy) 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`	The type of snapshot (legacy) processing that you want to continue if the snapshot (legacy) repository volume is full. You can choose to fail writes to the base volume (**`failBaseWrites`**) or fail writes to the snapshot (legacy) volume (**`failSnapshot`**). The default value is **`failSnapshot`**.

This example shows the **`create snapshotVolume`** command that includes user-defined parameters:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “create snapshotVolume baseVolume=\”Mars_Spirit_4\”
repositoryRAIDLevel=5 repositoryDriveCount=5
driveType=fibre userLabel=\”Mars_Spirit_4_snap1\”
repositoryUserLabel=\”Mars_Spirit_4rep1\”
warningThresholdPercent=75 repositoryPercentOfBase=40
repositoryFullPolicy=failSnapshot;”
```

This example shows how to use the command in a script file:

```
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;
```

Snapshot (Legacy) Volume Names and Snapshot (Legacy) Repository Volume Names

The snapshot (legacy) volume names and the snapshot (legacy) repository volume names can be any combination of alphanumeric characters, hyphens, and underscores. The maximum length of the volume names is 30 characters. You must enclose the volume name in double quotation marks. The character string cannot contain a new line. Make sure that you use unique names; if you do not use unique names, the controller firmware returns an error.

One technique for naming the snapshot (legacy) volume and the snapshot (legacy) repository volume is to add a hyphenated suffix to the original base volume name. The suffix distinguishes between the snapshot (legacy) volume and the snapshot (legacy) repository volume. For example, if you have a base volume with a name of Engineering Data, the snapshot (legacy) volume can have a name of Engineering Data-S1, and the snapshot (legacy) repository volume can have a name of Engineering Data-R1.

If you do not choose a unique name for either the snapshot (legacy) volume or the snapshot (legacy) repository volume, the controllers create a default name by using the base volume name. These examples are snapshot (legacy) volume names that the controllers might create:

- If the base volume name is aaa and does not have a snapshot (legacy) volume, the default snapshot (legacy) volume name is aaa1.
- If the base volume already has *n*-1 number of snapshot (legacy) volumes, the default name is aaa-*n*.
- If the base volume name is aaa and does not have a snapshot (legacy) repository volume, the default snapshot (legacy) repository volume name is aaa-R1.
- If the base volume already has *n*-1 number of snapshot (legacy) repository volumes, the default name is aaa-R*n*.

In the examples from the previous section, the user-defined snapshot (legacy) volume name was Mars_Spirit_4_snap1, and the user-defined snapshot (legacy) repository volume name was Mars_Spirit_4_rep1. The default name that was provided by the controller for the snapshot (legacy) volume was Mars_Spirit_4-1. The default name that was provided by the controller for the snapshot (legacy) repository volume was Mars_Spirit_4-R1.

Creating a Snapshot (Legacy) Schedule

You can create a snapshot (legacy) schedule in two ways:

- When you create a snapshot (legacy) volume using the **`create snapshotVolume`** command
- When you modify a snapshot (legacy) volume using the **`set (snapshot) volume`** command

The following table lists the parameters that you can use to set a schedule for a snapshot (legacy):

Table 20. Parameters for Creating a Snapshot (Legacy) Volume Schedule

Parameter	Description
`enableSchedule`	Use this parameter to turn on or to turn off the ability to schedule a snapshot (legacy) operation. To turn on snapshot (legacy) scheduling, set this parameter to **TRUE**. To turn off snapshot (legacy) scheduling, set this parameter to **FALSE**.
`schedule`	Use this parameter to schedule a snapshot (legacy) operation. You can use one of these options for setting a schedule for a snapshot (legacy) operation: ■ **`immediate`** ■ **`startDate`** ■ **`scheduleDay`** ■ **`startTime`** ■ **`scheduleInterval`** ■ **`endDate`** ■ **`noEndDate`** ■ **`timesPerDay`**
`rollbackPriority`	Use this parameter to determine whether system resources should be allocated to the rollback operation at the expense of system performance. A value of **`0`** indicates that the rollback operation is prioritized over all other host I/O. A value of **`4`** indicates that the rollback operation should be performed with minimal impact to host I/O. **NOTE** - This parameter is available only in the **`set (snapshot) volume`** command.

Scheduling Snapshots (Legacy)

Use the `enableSchedule` parameter and the `schedule` parameter to schedule automatic snapshots (legacy). Using these parameters, you can schedule snapshots (legacy) daily, weekly, or monthly (by day or by date). The `enableSchedule` parameter turns on or turns off the ability to schedule snapshots (legacy). When you enable scheduling, you use the `schedule` parameter to define when you want the snapshots (legacy) to occur.

This list explains how to use the options for the `schedule` parameter:

- **`immediate`** – As soon as you enter the command, a snapshot (legacy) volume is created, and a copy-on-write operation begins.
- **`startDate`** – A specific date on which you want to create a snapshot (legacy) volume and perform a copy-on-write operation. The format for entering the date is **`MM:DD:YY`**. If you do not provide a start date, the current date is used. An example of this option is `startDate=06:27:11`.
- **`scheduleDay`** - A day of the week on which you want to create a snapshot (legacy) volume and perform a copy-on-write operation. You can enter these values: **`monday`**, **`tuesday`**, **`wednesday`**, **`thursday`**, **`friday`**, **`saturday`**, **`sunday`**, and **`all`**. An example of this option is `scheduleDay=wednesday`.

- **`startTime`** – The time of a day that you want to create a snapshot (legacy) volume and start performing a copy-on-write operation. The format for entering the time is **`HH:MM`**, where **`HH`** is the hour and **`MM`** is the minute past the hour. Use a 24-hour clock. For example, 2:00 in the afternoon is 14:00. An example of this option is `startTime=14:27`.
- **`scheduleInterval`** – An amount of time, in minutes, that you want to have as a minimum between copy-on-write operation. You can possibly create a schedule in which you have overlapping copy-on-write operations because of the duration of a copy operation. You can make sure that you have time between copy-on-write operations by using this option. The maximum value for the `scheduleInterval` option is 1440 minutes. An example of this option is `scheduleInterval=180`.
- **`endDate`** – A specific date on which you want to stop creating a snapshot (legacy) volume and end the copy-on-write operation. The format for entering the date is **`MM:DD:YY`**. An example of this option is `endDate=11:26:11`.
- **`noEndDate`** – Use this option if you do not want your scheduled copy-on-write operation to end. If you later decide to end the copy-on-write operations you must re-enter the **`create snapshotVolume`** command and specify an end date.
- **`timesPerDay`** – The number of times that you want the schedule to run in a day. An example of this option is `timesPerDay=4`.

If you also use the `scheduleInterval` option, the firmware chooses between the `timesPerDay` option and the `scheduleInterval` option by selecting the lowest value of the two options. The firmware calculates an integer value for the `scheduleInterval` option by dividing 1440 by the `scheduleInterval` option value that you set. For example, 1440/180 = 8. The firmware then compares the `timesPerDay` integer value with the calculated `scheduleInterval` integer value and uses the smaller value.

To remove a schedule, use the **`delete snapshot (legacy)`** command with the `schedule` parameter. The **`delete snapshot (legacy)`** command with the `schedule` parameter deletes only the schedule, not the snapshot (legacy) volume.

Changing Snapshot (Legacy) Volume Settings

Use the **`set (snapshot) volume`** command to change these property settings for a snapshot (legacy) volume:

- The snapshot (legacy) volume name
- The warning threshold percent
- The snapshot (legacy) repository full policy
- The schedule options
- The snapshot (legacy) rollback priority

This example shows how to change a snapshot (legacy) volume name.

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “set volume [\”Mars_Spirit_4-1\”]
userLabel=\”Mars_Odyssey_3-2\”;”
```

This example shows how to use the command in a script file:

```
set volume [“Mars_Spirit_4-1”] userLabel=”Mars_Odyssey_3-2”;
```

When you change the warning threshold percent and the snapshot (legacy) repository full policy, you can apply the changes to one or several snapshot (legacy) volumes with this command. This example shows how to use the **`set (snapshot) volume`** command to change these properties on more than one snapshot (legacy) volume:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “set volume
[\”Mars_Spirit_4-1\” \”Mars_Spirit_4-2\” \”Mars_Spirit_4-3\”
warningThresholdPercent=50
repositoryFullPolicy=failBaseWrites;”
```

This example shows how to use the command in a script file:

```
set volume [“Mars_Spirit_4-1” “Mars_Spirit_4-2”
“Mars_Spirit_4-3”] warningThresholdPercent=50
repositoryFullPolicy=failBaseWrites;
```

Stopping, Restarting, and Deleting a Snapshot (Legacy) Volume

When you create a snapshot (legacy) volume, copy-on-write starts running immediately. As long as a snapshot (legacy) volume is enabled, storage array performance is impacted by the copy-on-write operations to the associated snapshot (legacy) repository volume.

If you no longer want copy-on-write operations to run, you can use the **`stop snapshot volume`** command to stop the copy-on-write operations. When you stop a snapshot (legacy) volume, the snapshot (legacy) volume and the snapshot (legacy) repository volume are still defined for the base volume. Only copy-on-write has stopped. This example shows how to stop a snapshot (legacy) volume:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “stop snapshot volumes
[\”Mars_Spirit_4-2\” \”Mars_Spirit_4-3\”];”
```

This example shows how to use the command in a script file:

```
stop snapshot volumes
[“Mars_Spirit_4-2” “Mars_Spirit_4-3”];
```

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

When you want to restart a copy-on-write operation, use the **`recreate snapshot volume`** command or the **`recreate snapshot collection`** command. The **`recreate snapshot volume`** command starts a fresh copy-on-write operation by using an existing snapshot (legacy) volume.

NOTE The snapshot (legacy) volume must be in either an Optimal state or a Disabled state.

When you restart a snapshot (legacy) volume, these actions occur:

- All copy-on-write data previously on the snapshot (legacy) repository volume is overwritten.
- Snapshot (Legacy) volume parameters and snapshot (legacy) repository volume parameters stay the same as the previously disabled snapshot (legacy) volume and the previously disabled snapshot (legacy) repository volume. You can also change the `userLabel` parameter, the `warningThresholdPercent` parameter, and the `repositoryFullPolicy` parameter when you restart the snapshot (legacy) volume.
- The original names for the snapshot (legacy) repository volume are retained.

This example shows how to restart a snapshot (legacy) volume:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
```

```
-c “recreate snapshot volumes
[\”Mars_Spirit_4-2\” \”Mars_Spirit_4-3\”];”
```

This example shows how to use the command in a script file:

```
recreate snapshot volumes
[“Mars_Spirit_4-2” “Mars_Spirit_4-3”];
```

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

Starting, Stopping, and Resuming a Snapshot (Legacy) Rollback

The snapshot (legacy) volume rollback commands provide a way for you to manage the content that is maintained by a snapshot (legacy) volume. A snapshot (legacy) rollback operation enables you to restore the contents of a base volume to a point-in-time image that was captured in a snapshot (legacy) volume. The base volume is then immediately accessible for read/write operation using the rolled-back content after the controller firmware accepts the rollback request. The read/write can take place while the rolled-back content is being transferred to the base volume. After the rollback operation is finished, the associated repository volume will be empty. The associated repository volume still has the capacity required to track new changes made by a host write operation after the point at which the rollback request was accepted.

Keep these guidelines in mind when managing a snapshot (legacy) volume rollback:

- A rollback operation copies from the repository volume only the data clusters that were changed between the time that the snapshot (legacy) was created and the time that the rollback was requested.
- Users can write to a snapshot (legacy) volume. Any changes made to a snapshot (legacy) volume after the creation of the original snapshot (legacy) image are included in the snapshot (legacy) rollback. Because of this, the rollback image to the base volume might not have the exact content as when the snapshot (legacy) image was originally created.
- The source snapshot (legacy) content is fully preserved and can be reused if you want to roll the snapshot (legacy) back again until you change the content of the snapshot (legacy) volume.
- You can perform only one snapshot (legacy) rollback operation at a time for a given base volume.
- While a rollback operation is taking place, you cannot delete the snapshot (legacy) volume.
- Rollback operations are mutually exclusive with volume copy operations that involve the base volume, either as the source or the destination of the volume copy. A request to start a rollback will be rejected if any volume copy relationship exists for the base volume. Similarly, an attempt to create a volume copy involving the base volume will be rejected if that base volume has a rollback operation in progress.
- Rollback operations involving the volumes in a Synchronous Mirroring relationship have these constraints:
  - If the base volume is acting as the secondary volume in a Synchronous Mirroring relationship, you cannot start a rollback operation.
  - If the base volume is acting as the primary volume in a Synchronous Mirroring relationship, you can start a rollback operation. In this case, the mirror relationship is forced into a Suspended state so that it immediately stops all updates from the primary storage array to the secondary storage array.
- A rollback operation can fail if the repository volume has any unreadable sectors that might have had repository data.

Setting Snapshot (Legacy) Rollback Priority

Rollback operations require some level of system overhead, which can reduce overall system performance. You can define the level of overhead the system devotes to a rollback operation using the **`set (snapshot) volume`** command. The **`set (snapshot) volume`** command has a `rollbackPriority` parameter. Values for the `rollbackPriority` parameter range from 0 through 4. A value of 0 means that the rollback operation has priority over all other host I/O. A value of 4 means that the rollback operation should be performed with minimal impact to host I/O.

Starting a Snapshot (Legacy) Rollback

When you start a snapshot (legacy) rollback, the contents of the base volume immediately start to change to the contents of the snapshot (legacy) volume. To start a snapshot (legacy) rollback use the **`start rollback`** command. This command accepts the name of one or more snapshot (legacy) volumes.

Stopping a Snapshot (Legacy) Rollback

ATTENTION Possible loss of data access – Stopping a snapshot (legacy) rollback can leave the base volume and snapshot volume unusable.

Stopping a snapshot (legacy) rollback leaves the base volume in an indeterminate state with potentially invalid or inconsistent data that is typically unusable by a host system. The related snapshot volume becomes disabled and unusable. Stop a snapshot (legacy) rollback only in cases where recovery options exist for restoring the data in a base volume. If you must stop a snapshot (legacy) rollback use the **`stop rollback`**.

Resuming a Snapshot (Legacy) Rollback

In some cases a rollback operation might pause because of a condition or action of the controller. If this occurs, you will see a status of Paused. After the controller is operating normally, you can resume a snapshot (legacy) rollback by using the **`resume rollback`** command.

Snapshot (Legacy) Rollback Status

You can see the status of a snapshot (legacy) rollback operation by running the **`show volume`** command on the snapshot (legacy) volume. The **`show volume`** command returns one of these statuses during a snapshot (legacy) rollback operation:

Table 21. Snapshot (Legacy) Rollback Operation Status

Status	Description
None	No snapshot (legacy) rollback operations are running.
In Progress	A snapshot (legacy) rollback operation is running. When a snapshot (legacy) rollback operation is running the amount of the rollback operation finished is shown as a percentage and an estimate of the time remaining is also shown.
Paused	A snapshot (legacy) rollback operation was started, but has been paused due to an error condition. If a snapshot (legacy) rollback operation has a status of Paused, the completion percentage shows the amount of work completed, and the estimated time until completion will be -1.

Status	Description
Pending	A snapshot (legacy) rollback operation request was accepted, but the rollback operation is currently waiting for previously scheduled snapshot (legacy) rollback operations to finish. The percentage complete is -1, and the estimated time until completion is -1.

Chapter 6 - Using the Snapshot Image Feature

A snapshot image is a logical image of the content of an associated base volume created at a specific moment. A snapshot image can be thought of as a restore point. A host cannot directly read from or write to the snapshot image because the snapshot image is used to save only the transient data captured from the base volume.

You must create a snapshot volume to enable host access to a complete copy of the data contained in the snapshot image. When the snapshot volume is a read/write volume, it has its own repository, which is used to save any subsequent modifications made by the host to the base volume without affecting the referenced snapshot image. If the snapshot volume is a read-only copy, it does not have a repository volume.

Snapshot images are useful any time you need to be able to roll back to a known good data set at a specific point in time. For example, before performing an operation on a volume that you might want to reverse, you can create a snapshot image to enable the reverse operation and restore the entire volume to its previous state. A snapshot image is created almost instantaneously, and initially uses no disk space, because it stores only the incremental changes needed to roll the volume back to the point-in-time when the snapshot image was created.

You can create snapshot images manually or automate the process with a schedule. With a schedule, a snapshot image is generated automatically, based on the date and time you choose, and is displayed with its associated snapshot group. You can create snapshot images for these storage objects:

- Standard volumes

  A standard volume has a fixed physical capacity that is entirely reserved for the data on the volume.
- Thin volumes

  A thin volume is dynamic. It has a small initial physical capacity and a large virtual capacity. The thin volume automatically expands to the virtual capacity as data is added to the volume.
- Consistency groups

  A consistency group is a container that holds several volumes so that you can manage all of the volumes as a single entity.

To create a snapshot image, you must first create a snapshot group and the associated snapshot repository volume. You can either use an existing repository volume or create a repository volume by performing these actions:

- Identifying the volume group or disk pool in which you want to place the repository volume
- The capacity for the repository volume

You can delete older snapshot images in a snapshot group. When a snapshot image is deleted, its definition is removed from the system, and the space occupied by the snapshot image in the repository is released and made available for reuse within the snapshot group.

You can roll back data by performing one of these actions:

- Creating a snapshot volume of a snapshot image, which enables you to retrieve deleted files from that snapshot volume (the base volume remains undisturbed).
- Restoring a snapshot image to the base volume, which enables you to roll back the base volume to a previous state.

Characteristics of Snapshot Images

- Snapshot images are always created inside snapshot groups.
- Each snapshot image is associated with exactly one snapshot group.
- There is a maximum limit of snapshot groups for a single associated base volume (depending on your configuration).

- There is a maximum limit of snapshot images per snapshot group (depending on your configuration).

Differences Between Snapshots (Legacy) and Snapshot Image Operations

A snapshot image is a logical point-in-time image of a volume. The snapshot image feature is similar to the snapshot (legacy) feature, with the following differences:

- The new snapshot image feature offers improved performance when a base volume has multiple point-in-time images. The snapshot (legacy) feature uses one data repository for each snapshot (legacy) volume. The new snapshot image feature uses one data repository for all of the snapshot images associated with a base volume. Therefore, when a base volume is written to, the new snapshot image feature requires only one write operation instead of multiple, sequential write operations.
- The new snapshot image feature adds the concept of a snapshot group. Because there is only one repository for multiple snapshot images, the repository is associated with the snapshot group instead of with the snapshot image as it is with the snapshot (legacy) feature.
- Unlike a snapshot (legacy) volume, a new snapshot image is not directly **`read-write`** accessible by hosts because the snapshot image is used only to save the changed data for a base volume. To provide hosts with **`read-write`** access to a snapshot image, you must first create a snapshot volume.
- You can create either snapshots (legacy) or snapshot images from a base volume, but not both.

You can create either a snapshot image that is capable of both reading operations and writing operations or you can create a read-only snapshot volume.

Snapshot Groups

A snapshot group is a collection of snapshot images of a single associated base volume. A snapshot image is always created within the context of a snapshot group, so the identity of the snapshot group is a key component that must be specified when creating a snapshot image.

A snapshot group maintains a sequence of snapshot images of a given base volume without impacting performance. You can set up a schedule for a snapshot group to automatically create a snapshot image at a specific time in the future or on a regular basis.

Each snapshot group has a strict ordering of snapshot images based on the time a snapshot image was created. These terms describe the time relationship of snapshot images:

- Successor – A snapshot image that is created after another snapshot image has been created, also called newest.
- Predecessor – A snapshot image that is created before another snapshot image, also called oldest.

The terms newest and oldest are arguments used in the CLI commands.

A snapshot group uses a repository volume to save all data for the snapshot images contained in the group. A snapshot image operation uses less disk space than a full physical copy because the data stored in the repository volume is only the data that has changed since the latest snapshot image.

A snapshot group is created initially with one repository volume. The repository volume initially contains a small amount of data, which increases with subsequent data updates. If needed, you can increase the size of the repository volume by combining unused standard volumes to the repository volume.

The only types of volumes that can contain a snapshot group are these:

- Standard volumes
- Thin volumes

A snapshot group cannot contain non-standard volumes, such as snapshot volumes. The base volume can reside on either a volume group or a disk pool.

NOTE Snapshot volumes and snapshot groups cannot coexist on the same base volume.

Characteristics of Snapshot Groups

- Snapshot groups can be initially created with or without snapshot images.
- Depending on your configuration, a single associated base volume has a maximum limit of snapshot groups.
- Depending on your configuration, a snapshot group has a maximum limit of snapshot images.
- Each snapshot image is associated with exactly one snapshot group.

Repository Volumes

Each snapshot image is created in the context of exactly one snapshot group. A snapshot group is a container of sequential snapshot images from a single associated base volume. A given snapshot group has exactly one repository volume that is used to save data for all of the snapshot images that are part of the snapshot group. When a new snapshot image is created in a snapshot group, the snapshot images of all previously-created snapshot images in that snapshot group become static and stop consuming additional repository space. The newly-created snapshot image begins consuming repository space during processing of ensuing copy-on-write actions for the associated base volume.

The repository volume is a standard volume that can be expanded to accommodate an increasing number of snapshot images. The repository volume is structured as a concatenated collection of up to 16 standard volumes. Initially, the repository volume has only a single standard volume, so its capacity is exactly that of its single standard volume. The repository volume is expanded by attaching additional standard volumes. The capacity then becomes the sum of the capacities of all of its concatenated volumes. Operating in this manner provides a flexible and low-overhead capacity expansion mechanism.

You can create a repository volume automatically using the default settings or you can manually create a repository volume by defining the capacity settings for the repository volume.

After the snapshot group and the associated repository volume are created, any write operation on the base volume necessitates that the affected content of the base volume be saved to a different location before it is overwritten by the new data. The snapshot image copy-on-write mechanism performs this task, and saves the required data to the repository volume. An index in the repository volume is then updated to reflect the new location of the snapshot image content for the affected logical block addressing (LBA) range. An ensuing base volume write to an LBA range that has already been preserved by a previous copy-on-write operation does not require an additional copy-on-write action, so the overhead of the copy is not repeated in such cases.

You can choose the way in which repository full conditions are managed through one of these modes of operation when such conditions arises:

- Auto-Purge Snapshot Images: Automatically delete the oldest snapshot images in the snapshot group to free up space that can be used to satisfy the copy-on-write operation capacity needs in the snapshot group repository.
- Fail Base Writes: Fail write requests to the base volume that triggered the repository access. In this case, the base volume, the snapshot group and its snapshot images and snapshot volumes remain in their previous states.

Each snapshot group can be configured independently in this regard. Expanding a repository volume does not directly change the state of any snapshot in the snapshot group. In particular, snapshot images in the purged state remain in that state even after a repository volume expansion creates additional free space for the snapshot group.

Repository Volume Names

The SANtricity GUI and CLI automatically provide a name (user label) every time a standard volume is created for use as a repository volume member. The user label string consists of the prefix “repos_” followed by a four-digit, zero-padded numeric suffix. The numeric value has a minimum value of 1, and is selected using the smallest previously-unused number among all existing repository volumes. Users cannot modify or override the name that is automatically provided by SANtricity.

NOTE SANtricity always uses a lower-case "repos_" prefix when it creates repository volumes. The CLI permits specification of an existing volume for use as a repository volume member only when that volume user label conforms to the pattern.

Snapshot Volumes

A snapshot volume provides the host access to the data contained in a snapshot image. A snapshot image cannot be directly read by a host because the snapshot image is only the changed data captured from the base volume. Therefore, you must create a volume that a host can access that is a merging of the data from the base volume and the snapshot image.

The snapshot volume can be designated as either **`read-only`** or **`read-write`**.

- A **`Read-Only`** snapshot volume provides a host application with **`READ`** access to a copy of the data contained in the snapshot image, but without the ability to modify the snapshot image. A **`Read-Only`** snapshot volume does not have an associated repository.
- A **`Read-Write`** snapshot volume requires an associated repository to provide the host application with **`WRITE`** access to a copy of the data contained in the snapshot image.

A snapshot volume that is designated as **`read-write`** must have its own repository volume to save any subsequent modifications made by the host application to the base volume without affecting the referenced snapshot image.

The snapshot is allocated from the storage pool from which the original snapshot image is allocated. All I/O write operations to the snapshot image are redirected to the snapshot volume repository that was allocated for saving data modifications. The data of the original snapshot image remains unchanged.

When you delete a snapshot volume, you can delete the snapshot repository volume or retain the snapshot repository volume as an unmapped volume.

Relationship Between Snapshot Images, Snapshot Groups, and Snapshot Volumes

The controller firmware and the SANtricity storage management software treats snapshot images, snapshot groups, and snapshot volumes as distinct entities relative to snapshots.

- A snapshot group has an association with a standard volume that is the base for the snapshot group.
- All snapshot images within a snapshot group have a direct association with that snapshot group.

- Each snapshot volume of a snapshot image has a direct association with that snapshot. In addition, each snapshot volume has a persistent relationship to the base volume of the snapshot image for which the snapshot volume was initially created. In other words, a snapshot volume is effectively “bound” to its base volume. Even if the snapshot volume is in a Stopped state, which detaches from a particular snapshot image. The snapshot volume retains its association with the base volume, and can be restarted on top of only the snapshot image for that base volume.
- The repository volumes for a snapshot group have an association with the snapshot group.

Consistency Groups

A consistency group is a container that holds several volumes so that you can manage all of the volumes as one entity. A consistency group enables users to take simultaneous snapshots of several volumes, thus ensuring consistent copies of a group of volumes.

If you frequently want to perform the same snapshot image operations on several volumes, you can create a consistency group to perform those operations. Any operation that you perform on the consistency group is performed simultaneously on all of the volumes in the consistency group. Some of the snapshot image operations that you can perform on a consistency group include creating, scheduling, and rolling back.

Each volume that belongs to a consistency group is referred to as a member volume. When you add a volume to a consistency group, the system automatically creates a new snapshot group that corresponds to this member volume. You can set up a schedule for a consistency group to automatically create a snapshot image of each member volume in the group at a specific time in the future or on a regular basis.

A consistency group pools several volumes together so that you can take a snapshot of all the volumes at the same point in time. This action creates a synchronized snapshot of all the volumes and is ideal for applications that span several volumes, for example, a database application that has the logs on one volume and the database on another volume.

You can use either of the following two methods to create a consistency group:

- Create the consistency group, and add the member volumes in one step.
- Create the consistency group, and then add the member volumes in a subsequent step.

Standard volumes and thin volumes are the only type of volumes that can be used for a consistency group. Non-standard volumes, such as snapshot (legacy) volumes, cannot be used for consistency groups. The base volume can reside on either a volume group or a disk pool.

Synchronous Mirroring and Snapshot Consistency Groups

- A volume can belong to several consistency groups. You can use consistency groups for snapshots and Synchronous Mirroring. Make sure to define separate and specific consistency groups for snapshots and Synchronous Mirroring.
- When you add a base volume that contains a consistency group to an asynchronous mirror group, the system automatically changes the repository full policy to automatically purge the oldest snapshot image and sets the auto-delete limit to the maximum allowable snapshot limit for a consistency group.
- All member volumes in a consistency group that also belong to an asynchronous mirror group, must belong to the same asynchronous mirror group.

Characteristics of Consistency Groups

A consistency group has the following characteristics:

- You can initially create consistency groups with or without member volumes.
- Depending on your configuration, a consistency group has a maximum allowable number of volumes.
- Depending on your configuration, a consistency group has a maximum allowable number of snapshot images.
- You can schedule a snapshot image to be created for a consistency group, which causes consistent snapshot images to be created for all member volumes.
- You can perform a Rollback operation for a consistency group.

Creating a Snapshot Group

Before you can create any snapshot images you must first create a snapshot group and the associated repository volume.

To create a new snapshot group use the **`create snapGroup`** command. This command creates a new snapshot group that is associated with a specific source volume. You also create a relationship to a repository volume by one of these methods:

- Use an existing repository volume
- Create a new repository volume in a volume group
- Create a new repository volume in a disk pool

To create a new repository volume, you identify either an existing volume group or an existing disk pool and define the size (capacity) for the repository volume. You define the size by entering either a percentage of the base volume or a specific size that you want to reserve for the repository volume.

This example shows a basic command in which a new snapshot group is being created and associated with an existing repository volume:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “create snapGroup userLabel=\”Data_Store_1\”
sourceVolume=\"Cont_Dev_04\"
repositoryVolume=\"repos_1234\";”
```

This command also has these additional parameters:

- Setting a repository full policy that defines whether you want to fail writes from the base volume or to delete (purge) snapshot images.
- The priority for rollback operations when you want to restore from the snapshot images. You can choose the amount of system processing to devote to rollback that ranges from a minimal impact to host I/O processing to a high impact that reduces host I/O processing.
- Warning limits for when the repository volume is approaching full. The limit is a percentage of the capacity of the repository volume.
- A minimum number of snapshot images that you want to delete if you have chosen to delete the snapshot images as the method for managing the repository full policy.
- Enabling and defining a schedule for capturing snapshot images.

Deleting a Snapshot Group

When you delete a snapshot group, the system performs the following actions when a snapshot group is deleted:

- Deletes all existing snapshot images from the snapshot group.
- Deletes the associated repository that exists for the snapshot group (if selected).
- Disables all the associated snapshot volumes that exist for the deleted snapshot images.

To delete the snapshot group, use this command:

```
delete snapGroup
```

If you want to retain the repository members, set the `deleteRepositoryMembers` parameter to **FALSE**.

Creating a Snapshot Image

To create a new snapshot image use the **`create snapImage`** command. This command creates a new snapshot image in one or more existing snapshot groups. Before you can create a snapshot image, you must first have at least one snapshot group into which you can place the snapshot image. To create a snapshot group use this command:

```
create snapGroup
```

The only parameters that you need to enter with this command are the names of the snapshot groups.

When you create a snapshot image of a consistency group, the result is a snapshot image of every member volume of the consistency group.

Canceling a Pending Snapshot Image

If you attempted to create the snapshot image in either a snapshot group or a consistency group, but the snapshot image was put in a Pending state, you can cancel the creation of the snapshot image. The snapshot image is in a Pending state due to the following concurrent conditions:

- The base volume for a snapshot group or one or more member volumes of a consistency group that contains this snapshot image is a member of an asynchronous mirror group.
- The volume or volumes are currently in a synchronizing operation.

The snapshot image creation operation completes as soon as the synchronization operation is complete. To cancel the pending snapshot image creation before the synchronization operation completes, use one of these commands:

- **`stop snapGroup pendingSnapImageCreation`**
- **`stop consistencyGroup pendingSnapImageCreation`**

When you cancel a pending snapshot image creation for a specific snapshot group or snapshot consistency group, only that group is disabled. All of the other groups stay in operation.

Creating a Snapshot Image Schedule

You can schedule creating regular snapshot images to enable file recovery and scheduled backups. You can create a schedule when you initially create a snapshot group or consistency group, or you can add one later to an existing snapshot group or consistency group. You can create a schedule that runs daily or weekly in which you select specific days of the week (Sunday through Saturday). You can temporarily suspend scheduled snapshot image creation by disabling the schedule.

- You can set up a schedule for a snapshot group to automatically create a snapshot image at a specific time in the future or on a regular basis.
- You can set up a schedule for a consistency group to automatically create a snapshot image of each member volume in the group at a specific time in the future or on a regular basis.

You can create a snapshot image schedule in two ways:

- When you create a snapshot group using the **`create snapGroup`** command
- When you create a snapshot consistency group using the **`create consistencyGroup`** command

The following table lists the parameters that you can use to set a schedule for creating a snapshot image.

Table 22. Parameters for Creating a Snapshot (Legacy) Volume Schedule

Parameter	Description
`enableSchedule`	Use this parameter to turn on or to turn off the ability to schedule a snapshot (legacy) operation. To turn on snapshot (legacy) scheduling, set this parameter to **TRUE**. To turn off snapshot (legacy) scheduling, set this parameter to **FALSE**.
`schedule`	Use this parameter to schedule a snapshot (legacy) operation. You can use one of these options for setting a schedule for a snapshot (legacy) operation: ■ **`immediate`** ■ **`startDate`** ■ **`scheduleDay`** ■ **`startTime`** ■ **`scheduleInterval`** ■ **`endDate`** ■ **`noEndDate`** ■ **`timesPerDay`**
`rollbackPriority`	Use this parameter to determine whether system resources should be allocated to the rollback operation at the expense of system performance. A value of **`0`** indicates that the rollback operation is prioritized over all other host I/O. A value of **`4`** indicates that the rollback operation should be performed with minimal impact to host I/O. **NOTE** - This parameter is available only in the **`set (snapshot) volume`** command.

Scheduling Snapshot Images

Use the `enableSchedule` parameter and the `schedule` parameter to schedule automatically creating snapshot images. Using these parameters, you can schedule snapshot images daily, weekly, or monthly (by day or by date). The `enableSchedule` parameter turns on or turns off the ability to schedule snapshot images. When you enable scheduling, you use the `schedule` parameter to define when you want the snapshot images to occur.

This list explains how to use the options for the `schedule` parameter:

- **`immediate`** – As soon as you enter the command, a snapshot image is created, and a copy-on-write operation begins.
- **`startDate`** – A specific date on which you want to create a snapshot image and perform a copy-on-write operation. The format for entering the date is **`MM:DD:YY`**. If you do not provide a start date, the current date is used. An example of this option is `startDate=06:27:11`.
- **`scheduleDay`** – A day of the week on which you want to create a snapshot image and perform a copy-on-write operation. You can enter these values: **`monday`**, **`tuesday`**, **`wednesday`**, **`thursday`**, **`friday`**, **`saturday`**, **`sunday`**, and **`all`**. An example of this option is `scheduleDay=wednesday`.
- **`startTime`** – The time of a day that you want to create a snapshot image and start performing a copy-on-write operation. The format for entering the time is **`HH:MM`**, where **`HH`** is the hour and **`MM`** is the minute past the hour. Use a 24-hour clock. For example, 2:00 in the afternoon is 14:00. An example of this option is `startTime=14:27`.
- **`scheduleInterval`** – An amount of time, in minutes, that you want to have as a minimum between copy-on-write operation. You can possibly create a schedule in which you have overlapping copy-on-write operations because of the duration of a copy operation. You can make sure that you have time between copy-on-write operations by using this option. The maximum value for the `scheduleInterval` option is 1440 minutes. An example of this option is `scheduleInterval=180`.
- **`endDate`** – A specific date on which you want to stop creating a snapshot image and end the copy-on-write operation. The format for entering the date is **`MM:DD:YY`**. An example of this option is `endDate=11:26:11`.
- **`noEndDate`** – Use this option if you do not want your scheduled copy-on-write operation to end. If you later decide to end the copy-on-write operations you must re-enter the **`create snapGroup`** command or the **`create consistencyGroup`** command and specify an end date.
- **`timesPerDay`** – The number of times that you want the schedule to run in a day. An example of this option is `timesPerDay=4`.

If you also use the `scheduleInterval` option, the firmware chooses between the `timesPerDay` option and the `scheduleInterval` option by selecting the lowest value of the two options. The firmware calculates an integer value for the `scheduleInterval` option by dividing 1440 by the `scheduleInterval` option value that you set. For example, 1440/180 = 8. The firmware then compares the `timesPerDay` integer value with the calculated `scheduleInterval` integer value and uses the smaller value.

To remove a schedule, use the **`create snapGroup`** command or the **`create consistencyGroup`** command and set the `enableSchedule` parameter to **`FALSE`**. The `enableSchedule` parameter set to **`FALSE`** turns off the schedule, but not creating snapshot images.

Deleting a Snapshot Group

When you delete a snapshot image from a snapshot group, the system performs the following actions:

- Deletes the snapshot image from the storage array
- Releases the repository's reserve space for reuse within the snapshot group
- Disables all the associated snapshot volumes that exist for the deleted snapshot image

For a consistency group you can delete:

- A single snapshot image
- Multiple snapshot images that have the same sequence number and creation timestamp

When a snapshot image(s) is deleted from a consistency group, the system performs the following actions:

- Deletes the snapshot image from the storage array
- Releases the repository's reserve space for reuse within the consistency group
- Moves any member volume, associated with the deleted snapshot image(s), to a Stopped state
- Disables the member snapshot volumes associated with the deleted snapshot image(s)

To delete the snapshot image, use this command:

```
delete snapImage
```

Optionally you can choose to keep a number of snapshot images with these parameters:

- `deleteCount` – This parameter deletes the oldest snapshot image first and continues to delete the oldest snapshot images until reaching the number that you enter. If the number that you enter is greater than the number of snapshot images, then all of the snapshot images are deleted.
- `retainCount` – This parameter keeps the most recent snapshot images in the consistency group.

If you have a snapshot volume associated with the associated consistency group you can choose to keep or delete the snapshot volume. To keep the snapshot volume, set the `ignoreSnapVolume` parameter to **TRUE**. To delete the snapshot volume, set the `ignoreSnapVolume` parameter to **FALSE**. The default is **FALSE**. This parameter applies only if the consistency group snapshot image is associated with a consistency group snapshot volume.

Creating a Snapshot Consistency Group

A consistency group snapshot volume combines several snapshot volumes to provide host access to a snapshot image that has been taken for each selected member volume at the same moment in time.

The consistency group snapshot volume can be designated as either **`read-only`** or **`read-write.`** Read-write consistency group snapshot volumes require a repository for each member volume that you select to save any subsequent modifications made by the host application to the base volume without affecting the referenced snapshot image. Each member repository is created at the same time the consistency group snapshot volume is created.

NOTE A **`read-write`** snapshot volume requires an overall repository. The overall repository is created initially with one individual repository volume. You can later add additional volumes to the overall repository to expand the overall repository capacity.

Prerequisites

- The Snapshot premium feature must be enabled on the storage array.
- The consistency group must contain at least one member volume before you can create a consistency group snapshot volume.

Guidelines

Keep these guidelines in mind when creating a consistency group snapshot volume:

- There is a maximum allowable limit to the number of snapshot images for a consistency group (depending on your configuration).
- You cannot create a snapshot volume of a failed volume.
- You can change the size of the snapshot repository. If you have the storage capacity you can increase the size of the snapshot repository to avoid a repository full message. Conversely, if you find that the snapshot volume repository is larger than you need, you can reduce its size to free up space that is needed by other logical volumes.
- Both of these conditions together might cause the creation of a snapshot image to enter in a Pending state when you try to create a snapshot volume:
  - The base volume that contains this snapshot image is a member of an asynchronous mirror group
  - The base volume is currently in a synchronization operation. As soon as the synchronization operation finishes, the snapshot image is created.

To create a consistency group snapshot volume, use this command:

```
create snapVolume
```

When you enter this command you must give the snapshot volume a unique name and identify the snapshot image to be associated with the snapshot volume.

Optionally you can choose the name of the repository volume associated with the snapshot volume and set a warning limit for that repository volume.

Deleting a Snapshot Consistency Group

When you delete a snapshot consistency group, the system performs the following actions:

- Deletes all existing snapshot images from the consistency group.
- Deletes all existing snapshot volumes from the consistency group.
- Deletes all the associated snapshot images that exist for each member volume in the consistency group.
- Deletes all the associated snapshot volumes that exist for each member volume in the consistency group.
- Deletes all associated repositories that exist for each member volume in the consistency group (if selected).

To delete the snapshot consistency group, use this command:

```
delete cgSnapImage consistencyGroup
```

Optionally you can choose to keep a number of snapshot images with these parameters:

- `deleteCount` – This parameter deletes the oldest snapshot image first and continues to delete the oldest snapshot images until reaching the number that you enter. If the number that you enter is greater than the number of snapshot images, all of the snapshot images are deleted.
- `retainCount` – This parameter keeps the most recent snapshot images in the consistency group.

If you have a snapshot volume associated with the associated consistency group you can choose to keep or delete the snapshot volume. To keep the snapshot volume set the `ignoreSnapVolume` parameter to **TRUE**. To delete the snapshot volume set the `ignoreSnapVolume` parameter to **FALSE**. The default is **FALSE**. This parameter applies only if the consistency group snapshot image is associated with a consistency group snapshot volume.

Creating a Snapshot Volume

You create a snapshot volume to provide host access to a snapshot image within a snapshot group. A **read-write** snapshot volume has its own repository that is used to save any subsequent modifications made by the host application to the base volume without affecting the referenced snapshot image.

You create a snapshot volume to provide host access to a snapshot image within a snapshot group. A **read-write** snapshot volume has its own repository that is used to save any subsequent modifications made by the host application to the base volume without affecting the referenced snapshot image.

The snapshot volume can be designated as either **read-only** or **read-write**.

- A **read-only** snapshot volume provides a host application with **READ** access to a copy of the data contained in the snapshot image, but without the ability to modify the snapshot image. A **read-only** snapshot volume does not have an associated repository.
- A **read-write** snapshot volume requires an associated repository to provide the host application with **WRITE** access to a copy of the data contained in the snapshot image.

Prerequisites

- The Snapshot premium feature must be enabled on the local storage arrays.

Guidelines

Keep these guidelines in mind when creating a snapshot volume:

- You cannot create a snapshot volume of a Failed base volume.
- You can change the size of the snapshot repository. If you have the storage capacity you can increase the size of the snapshot repository to avoid a repository full message. Conversely, if you find that the snapshot volume repository is larger than you need, you can reduce its size to free up space that is needed by other logical volumes.
- Both of these conditions together might cause the creation of a snapshot image to enter in a Pending state when you try to create a snapshot volume:
  - The base volume that contains this snapshot image is a member of an asynchronous mirror group
  - The base volume is currently in a synchronization operation. As soon as the synchronization operation finishes, the snapshot image is created.

Creating a Snapshot Volume

To create a **read-only** snapshot volume, use this command:

```
create snapVolume readOnly
```

To create a **read-write** snapshot volume, use this command:

```
create snapVolume
```

Resuming a Consistency Group Snapshot Volume

If a snapshot volume has entered a Stopped state, the snapshot volume becomes inaccessible for read or write operations. The snapshot volume also is detached from the snapshot with which it was previously associated. The snapshot volume still remains logically bound to the associated base volume.

To restart a snapshot volume that has entered a Stopped state, use this command:

```
resume cgSnapVolume
```

When you run this command, you must identify the snapshot volume and the consistency group snapshot image that were being used when the snapshot volume stopped.

Deleting a Snapshot Volume

If you no longer need a snapshot volume, you can delete the volume using this command:

```
delete snapVolume
```

When you delete a snapshot volume the system performs the following actions:

- Deletes all existing links with the snapshot group
- Deletes the associated repository that exists for the snapshot group (if selected).

If you want to retain the repository members, set the `deleteRepositoryMembers` parameter to **FALSE**.

Changing the Size of a Repository Volume

You can increase or decrease the size of a repository volume.

Increasing the Size of a Repository Volume

Because a repository volume is comprised of one or more standard volumes, you can increase the storage capacity of an existing repository for these storage objects:

- Snapshot group
- Snapshot volume
- Consistency group member volume
- Consistency group member snapshot volume

Typically, you increase capacity when you receive a warning that the repository is becoming full. You can increase the repository capacity by performing one of these tasks:

- Adding one or more existing repository volumes
- Creating a new repository volume using free capacity that is available on a volume group or disk pool

If any volume group or disk pool does not have free capacity, you can add unconfigured capacity in the form of unused drives to a volume group or disk pool.

Prerequisites

You cannot increase the storage capacity of a repository volume if one of these conditions exists:

- The volume that you want to add does not have an Optimal status.
- Any volume in the volume group or the disk pool that you want to add is in any state of modification.

- The volume group or the disk pool does not have any free capacity.
- The volume group or the disk pool does not have any unconfigured capacity.
- Eligible volumes are not available.

Guidelines

- Make sure that a base volume and each of the individual volumes in the overall repository have the same Quality of Service (QoS) attributes, specifically for the following characteristics:
  - **RAID Level:** A repository in a disk pool is considered to have a matching RAID level for any base volume on a volume group, regardless of the base volume's actual RAID level. However, a repository on a volume group is considered to have a matching RAID level only if that RAID level is identical to the RAID level of the base volume.
  - **Drive Type:** A match requires that the base volume and the repository volume reside on either a volume group or disk pool with identical drive type attributes.
- You cannot increase or decrease the repository capacity for a snapshot volume that is **`read-only`** because it does not have an associated repository. Only snapshot volumes that are **`read-write`** require a repository.

To increase the size of a repository volume use one of these commands:

- `set snapVolume ["snapVolumeName"] increaseRepositoryCapacity repositoryVolumes=(repos_xxxx)`
- `set snapGroup ["snapGroupName"] increaseRepositoryCapacity repositoryVolumes=(repos_xxxx)`

Decreasing the Size of a Repository Volume

If a repository volume has more capacity than is needed, you can reduce the capacity of the repository volume by removing member volumes from the end of its concatenated set. A repository volume always must retain at least one member after such a reduction. The standard volumes that are removed in such an operation are effectively "detached" from the repository volume, thus reducing the repository volume capacity, and making the detached volumes into standard volumes. The detached volumes can then be reused to provide additional capacity for the same or a different repository volume.

You can reduce the storage capacity of an existing repository volume for the following storage objects:

- Snapshot group
- Snapshot volume
- Consistency group member volume
- Consistency group member snapshot volume

Prerequisites

You cannot decrease the storage capacity of the overall repository if one of these conditions exists:

- The overall repository contains only one repository member volume.
- If there are one or more snapshot images associated with the overall repository.
- If a snapshot volume or a consistency group member snapshot volume is disabled.

Guidelines

- You can remove repository member volumes only in the reverse order that they were added.
- An overall repository must have at least one repository member volume.

- You cannot increase or decrease the repository capacity for a snapshot volume that is read-only because it does not have an associated repository. Only snapshot volumes that are read-write require a repository.
- When you decrease capacity for a snapshot volume or a consistency group member snapshot volume, the system automatically transitions the volume to a Disabled state.

To decrease the size of a repository volume use one of these commands:

- `set snapVolume ["snapVolumeName"] decreaseRepositoryCapacity count=numberOfVolumes`
- `set snapGroup ["snapGroupName"] decreaseRepositoryCapacity count=numberOfVolumes`

Starting, Stopping, and Resuming a Snapshot Image Rollback

Snapshot images are useful any time you need to be able to roll back to a known good data set at a specific point in time. For example, before performing a risky operation on a volume, you can create a snapshot image to enable "undo" capability for the entire volume. You can start a rollback from the following types of snapshot images:

- Snapshot image of a base volume, which allows you to roll back the base volume associated with a snapshot group to a previous state.
- Consistency group snapshot image, which allows you to roll back all or select member volumes of the consistency group to a previous state.

The snapshot image rollback commands provide a way for you to manage the content that is maintained by snapshot groups and snapshot consistency groups. A snapshot image rollback operation enables you to restore the contents of a base volume to a point-in-time image that was captured when a snapshot image was created. The base volume is then immediately accessible for read/write operation using the rolled-back content after the controller firmware accepts the rollback request. The read/write operation can take place while the rolled-back content is being transferred to the base volume.

You can roll back data by performing one of these operations:

- Creating a snapshot volume of a snapshot image, which allows you to retrieve deleted files from that snapshot volume (the base volume remains undisturbed).
- Restoring a snapshot image to the base volume, which allows you to roll back the base volume to a previous point-in-time.

NOTE The host has immediate access to the newly rolled-back base volume, but the existing base volume does not allow the host **`read-write`** access after the rollback is initiated. You can create a snapshot of the base volume just before initiating the rollback to preserve the pre-rollback base volume for recovery purposes.

Keep these guidelines in mind before you start a rollback operation:

- The rollback operation does not change the content of the snapshot images that are associated with the base volume.
- You cannot perform the following actions when a rollback operation is in progress:
  - Delete the snapshot image that is being used for the rollback.
  - Create a new snapshot image for a base volume that is participating in a rollback operation.
  - Change the associated snapshot group's Repository-Full Policy.
- You cannot start a rollback operation when any of these operations are in progress in the storage array:
  - Dynamic Capacity Expansion (DCE) to increase the capacity of a volume group.

  - Dynamic Volume Expansion (DVE) to increase the capacity of a volume.
  - Dynamic RAID Migration (DRM) to change the RAID level of a volume group.
  - Dynamic Segment Size (DSS) to change the segment size of a volume.
- You cannot start a rollback operation if the base volume is participating in a volume copy.
- You cannot start a rollback operation if the base volume is a secondary volume in a remote mirror. However, if the base volume is the primary volume in a remote mirror, you can start a rollback operation. Additionally, you cannot perform a role reversal in a remote mirror if the primary volume is participating in a rollback operation.
- A rollback operation fails if any of the used capacity in the associated snapshot repository volume has unreadable sectors.

Setting Snapshot Image Rollback Priority

Rollback operations require some level of system overhead, which can reduce overall system performance. You can define the level of overhead the system devotes to a rollback operation using the **`create snapGroup`** command or the **`create consistencyGroup`** command. These commands have a `rollbackPriority` parameter. Values for the `rollbackPriority` parameter range from `highest` through `lowest`. The `highest` value means that the rollback operation has priority over all other host I/O. The `lowest` value means that the rollback operation should be performed with minimal impact to host I/O.

Starting a Snapshot Image Rollback

When you start a snapshot image rollback, the contents of the base volume immediately start to change to the contents of the snapshot image. To start a snapshot image rollback use the **`start snapImage rollback`** command or the **`start cgSnapImagerollback`** command. These commands accept the name of one or more member volumes. The **`start snapImage rollback`** works with specific snapshot images. The **`start cgSnapImagerollback`** command works with specific member volumes in the consistency group.

Stopping a Snapshot Image Rollback

ATTENTION Possible loss of data access – Stopping a snapshot image rollback can leave the base volume and the snapshot image unusable.

Stopping a snapshot image rollback leaves the base volume in an indeterminate state with potentially invalid or inconsistent data that is typically unusable by a host system. The base volume appears as failed in the storage management software. Stop a snapshot image rollback only in cases where recovery options exist for restoring the data in a base volume. If you must stop a snapshot image rollback, use the **`stop snapImage rollback`** command or the **`stop cgSnapImage rollback`** command.

You can cancel an active rollback that is in progress (actively copying data), a pending rollback (in a pending queue awaiting resources to start), or a rollback that has been paused due to an error.

After you cancel a rollback operation, you must take one of the following actions:

- Reinitialize the content of the base volume.
- Perform a new rollback operation to restore the base volume (using either the same snapshot image that was used in the **Cancel Rollback** operation or a different snapshot image to perform the new rollback operation).

NOTE If the snapshot group on which the snapshot image resides has one or more snapshot images that have been automatically purged, the snapshot image used for the rollback operation might not be available for future rollbacks.

Resuming a Snapshot Image Rollback

In some cases a rollback operation might pause because of a condition or action of the controller. If this occurs, you see a status of Paused. After the controller is operating normally, you can resume a snapshot image rollback by using either the **`resume snapImage rollback`** command or the **`resume cgSnapVolume`** command.

Snapshot Image Rollback Status

You can see the status of a snapshot image rollback operation by running either the **`show snapImage`** command or the **`show cgSnapImage`** command. The following table shows the statuses that these commands return during a snapshot image rollback.

Table 23. Snapshot (Legacy) Rollback Operation Status

Status	Description
None	No snapshot image rollback operations are running.
In Progress	A snapshot image rollback operation is running. When a snapshot image rollback operation is running, the amount of the rollback operation finished is shown as a percentage and an estimate of the time remaining is also shown.
Paused	A snapshot image rollback operation was started but has been paused due to an error condition. If a snapshot image rollback operation has a status of Paused, the completion percentage shows the amount of work completed, and the estimated time until completion is -1.
Pending	A snapshot image rollback operation request was accepted, but the rollback operation is currently waiting for previously scheduled snapshot image rollback operations to finish. The percentage complete is -1, and the estimated time until completion is -1.

Chapter 7 - Using the Asynchronous Mirroring Feature

The Asynchronous Mirroring premium feature replicates the data between storage arrays over a remote distance. In the event of a disaster or a catastrophic failure on one storage array, you can promote the second storage array to take over responsibility for computing services. Asynchronous mirroring is designed for extended storage environments in which the storage arrays that are used for asynchronous mirroring are maintained at separate sites. Volumes on one storage array are mirrored to volumes on another storage array across a fabric SAN. Because asynchronous mirroring is storage based, it does not require any server overhead or application overhead.

Asynchronous mirroring is managed on a per-volume basis, enabling you to associate a distinct remote mirrored volume with one or more primary volume on a given storage array. Point-in-time images are used on the primary volume and the secondary volume to batch the resynchronization process. A data repository volume is required for each mirrored volume. Also, you can create asynchronous mirror groups that enable you to manage the synchronization process as a set to create consistent data on the remote storage array.

Asynchronous mirroring can use either Fibre Channel or iSCSI to communicate with the remote array. Inter-controller communication for asynchronous mirroring uses the host-connected ports to initiate connections to the remote system. The iSCSI inter-controller communication must use a host-connect port, not the management Ethernet port. A controller can receive host I/O through one protocol and use a different protocol for asynchronous mirroring with a remote storage array. For example, a host can be attached to the controller through a SAS connection while the inter-controller mirroring data is sent over an iSCSI connection to the remote storage array.

The asynchronous write mode of the legacy Synchronous Mirroring premium feature is still supported, but the new Asynchronous Mirroring premium feature is the preferred method to use if available on your storage array. Some key differences between Asynchronous Mirroring premium feature and the Synchronous Mirroring premium feature are these:

- With the Asynchronous Mirroring premium feature, a data repository volume is required for each mirror. The Synchronous Mirroring premium feature has a single repository for all mirrored volumes.
- The Asynchronous Mirroring premium feature uses point-in-time images on the primary and secondary volumes to batch the resynchronization process.
- The Asynchronous Mirroring premium feature organizes mirrored volumes into groups to manage the synchronization process to create a consistent data set on the remote storage array.
- The Asynchronous Mirroring premium feature supports iSCSI and Fibre Channel connections between storage arrays. The Synchronous Mirroring premium feature supports only Fibre Channel connections.

You can use asynchronous mirroring for these functions:

- **Disaster recovery** – You can replicate data from one site to another site, which provides an exact duplicate at the remote (secondary) site. If the primary site fails, you can use mirrored data at the remote site for failover and recovery. You can then shift storage operations to the remote site for continued operation of all of the services that are usually provided by the primary site.
- **Data vaulting and data availability** – You can send data off site where it can be protected. You can then use the offset 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** – You can have two storage arrays back up each other by duplicating critical volumes on each storage array to volumes on the other storage array. This action lets each storage array recover data from the other storage array in the event of any service interruptions.

How Asynchronous Mirroring Works

When you set up asynchronous mirroring, you create a one-to-one relationship between the volumes in a remote-mirrored pair. The remote-mirrored pair consists of a primary volume on a local storage array and a secondary volume on a storage array at another site. The primary-secondary role is defined in an asynchronous mirror group. Mirrored relationships are created by adding a volume to the primary asynchronous mirror group and adding the corresponding volume to the secondary asynchronous mirror group on the remote storage array. The maximum numbers of supported asynchronous mirrored pairs are listed in the following table.

NOTE Remember, most mirror operations (such as creating relationships, synchronization settings, and mirroring roles) are managed through an asynchronous mirror group.

Table 24. Maximum Number of Defined Mirrors

Base System	Per Storage Array	Per Asynchronous Mirroring Group
QD6000	128	64

Note: The Asynchronous Mirroring premium feature is supported only on QD6000 configurations.

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 controller owner of the primary volume but does not accept host write requests. Hosts are able to read from the secondary volume, which appears as read-only.

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

When write operations are performed to the primary volume of an asynchronous mirrored pair, the modified data region of the primary volume is tracked. Periodically, the firmware creates a new point-in-time image of the primary volume and sends the changed data regions to the secondary volume. When data synchronization completes, the system uses the point-in-time images on the secondary volume to ensure that the data is maintained in a consistent state during subsequent synchronization operations to the secondary volume.

The automatic cycle in an active asynchronous mirroring session works as follows to maintain data at a remote site to be a point-in-time consistent copy of data at the local site.

1. Asynchronous mirror groups consisting of volumes are created at the local site.
2. Increments of consistent data are sent to the remote site.
3. Point-in-time copy operations are performed at the remote site.

These steps are repeated according to the defined mirroring intervals.

The fundamental assumptions of the Asynchronous Mirroring premium feature are these:

- The data link between the primary storage array and the secondary storage array has significantly lower bandwidth and higher latency than the host interface. For this reason, the mirrored data movement must be decoupled from the primary host data requests to minimize the performance impact to the host application. Additionally, with a restricted data link unsynchronized regions between the primary and secondary volumes must be tracked in sufficiently small granularity to minimize data movement. Long distance data links to remote storage arrays require additional networking equipment, which can result in varying throughput performance.
- Data on the secondary volume must support a site-level failover for disaster recovery. For this reason, the data on the secondary volume is protected during the synchronization process so that writes to the secondary volume do not render the volume data unusable. Additionally, many applications require the use of more than one volume, each of which must be mirrored in order to support a site-level failover. In these cases, the set of volumes must be mirrored as a set, and the synchronization process must coordinate the data movement and synchronization intervals to create a consistent, usable data set on the secondary storage array.

Configurating for Asynchronous Mirroring

Asynchronous mirroring operations are performed between controllers that have the same IDs on the primary and secondary storage arrays; that is, controller A on the primary storage array interacts only with controller A on the secondary storage array and controller B on the primary storage array interacts only with controller B on the secondary storage array. Asynchronous mirroring operations are not attempted between controllers that have different IDs. The network environment does not need to provide connectivity between the controllers that have different IDs; however, it is likely to be more cost effective to use a fabric/switch configuration that does provide such connectivity, even though the firmware does not require it.

For Fibre Channel topology the Asynchronous Mirroring premium feature requires that one host-side Fibre Channel port of each controller be dedicated to mirroring operations. An additional requirement for connectivity is that the dedicated asynchronous mirroring ports must be attached to a Fibre Channel Fabric. The dedicated port is activated by the storage system administrator. Activating the dedicated port is a different operation than enabling the premium feature. After you activate the dedicated port you must enable the Asynchronous Mirroring premium feature. Enabling the Asynchronous Mirroring premium feature performs these actions:

- Activates the storage array for Fibre Channel mirroring
- Enables the controller firmware to support your creation of mirror groups and mirror pairs

When mirror groups or mirror pairs exist, you cannot deactivate the Asynchronous Mirroring premium feature.

An iSCSI topology does not require a dedicated port as with Fibre Channel. The activation step is not required when setting up asynchronous mirroring over iSCSI. The controller firmware maintains a list of remote storage arrays that the iSCSI initiator attempts to establish sessions using all portals of the portal group. The first portal that successfully establishes an iSCSI connection is used for all subsequent communication with that remote storage array. If communication fails, a new session is attempted using all portals in the portal group. iSCSI ports are configured at the system level on a port-by-port basis. For configuration messaging and data transfer, the communication between the controller uses these global settings:

- VLAN – Both local and remote systems must have the same VLAN setting in order to communicate
- iSCSI listening port
- Jumbo Frames
- Ethernet Priority

Asynchronous Mirror Groups

Some applications, such as file systems and databases, distribute data storage across many volumes. To create a failover site for such applications, all of the application data volumes must be replicated on a storage array at a remote site. The synchronization between the original data volumes and the replicated data volumes must be coordinated so that writes on the local storage array are accurately reflected on the remote storage array. The purpose of an asynchronous mirror group is to associate a set of volumes so that mirror synchronization can be coordinated for the volume set, creating a consistent data set on the remote storage system.

- The local storage array is the primary side of the asynchronous mirror group. The remote storage array is the secondary side of the asynchronous mirror group.
- All volumes added to the asynchronous mirror group on the local storage array hold the primary role in the mirrored relationship.
- All volumes added to the asynchronous mirror group on the remote storage array hold the secondary role in the mirrored relationship.

During the synchronization, point-in-time images for all primary volumes are created at the same time. Data is replicated for all volumes in the group to the remote system within the same synchronization interval. The synchronization progress for a given resynchronization operation varies among member volumes. However, point-in-time images of the Secondary volumes at the completion of data synchronization are created at the same time after all members of the group have completed the synchronization. If resynchronization fails (even due to just one member), prior consistent images remain on the Secondary AMG to preserve the last known consistent data.

The properties associated with the asynchronous mirror group include the following:

- World Wide Name
- User label
- Role: primary or secondary

NOTE The role is an asynchronous mirror group attribute, not a mirrored volume attribute. Role reversal affects all volumes in the asynchronous mirror group.

- Mirrored I/O Channel Type: Fibre Channel or iSCSI
- Timestamp of the last known recovery point
- Resynchronization interval

  The resynchronization interval is the amount of time between automatically sending updates of modified data from the primary storage array to the secondary storage array. The interval, expressed in minutes, represents the time between the starting points of sending updates from the primary to the secondary. A resynchronization interval of zero means that synchronization is manual.
- Synchronization progress

  The synchronization progress is the percent complete and an estimated time to completion of the current synchronization activity if in progress.
- Alert threshold for resynchronization completion time (for example, alert if the resynchronization takes too long)
- Alert threshold for recovery point degradation (such as, age of the Secondary PiTs)
- Repository utilization warning threshold
- Member volume list

The following properties are shared between the local storage array and remote storage array:

- World Wide Name
- User label

- Mirrored I/O Channel Type: Fibre Channel or iSCSI
- Resynchronization interval
- Alert threshold for resynchronization completion time (for example, alert if the resynchronization takes too long)
- Alert threshold for recovery point degradation (such as, age of the Secondary PiTs)
- Repository utilization warning threshold

The shared properties are communicated to the remote storage array when you create the asynchronous mirror group. Changes to a shared property are made first on the local storage array then communicated to the remote storage array. If a shared property fails to change on both the local storage array and the remote storage array, the local storage array is considered to be the master setting. When the two storage arrays reconnect after a communication outage or reboot, the asynchronous mirror group properties are reconciled from the local storage array.

If the two storage arrays detect a dual role conflict after a communication outage is resolved, the asynchronous mirror group properties are not reconciled until the dual-role conflict is resolved. To resolve the role conflict you must determine which storage array is the primary storage array. At that time, the asynchronous mirror group properties are then assumed from the side that you have defined to be the primary side.

Mirror Repository Volumes

A mirror repository volume is a special volume in the storage array that is created as a resource for the controller owner of the primary volume in a remote mirrored pair. The controller stores mirror 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 the accidental powering down of the storage arrays.

Mirror repository volumes are used to manage mirror data synchronization. Mirror repository volumes are required for both the primary and secondary volumes in a mirror pair. This mirror repository volume management is different from the Synchronous Mirroring premium feature where a single mirror repository is set up once and then used for all Synchronous Mirroring relationships.

The mirror repository pair stores three types of data:

- Copy-on-write repository data that is used for preserving resynchronization images on the mirror-primary and recovery point images on the mirror-secondary. The copy-on-write repository is structurally similar to the repository use for snapshot groups.
- A pair of delta logs that are used to track regions of the mirror-primary volume that are written between synchronization intervals. Even though the delta logs are only used on the primary side of the mirror, they are also allocated on the secondary side to support a role-reversal.
- A log that tracks synchronization statistics on each mirror pair.

Asynchronous mirroring repository volumes are expandable repository volumes (ERV). The minimum mirror repository size is the greater of either:

- 0.02 percent of the base volume capacity (regardless of role)
- 32 MB

You can expand the capacity of the mirror repository volumes using the standard rules for ERV expansion. You cannot reduce the size of the mirror repository volumes.

The maximum repository capacity is 101 percent of the base volume capacity. Primary and secondary mirror repositories are not required to be the same size. Mirror repository volumes are independent of the associated

primary mirror or secondary mirror volume in that they can be created on separate volume groups with different RAID levels. Mirror repository volumes must have compatible security and data assurance quality of service as the associated mirror volume. For example, if a mirror volume has data security enabled, the associated mirror repository must have data security enabled. If the mirror volume has data assurance enabled, the associated mirror repository volume must have data assurance enabled.

Because of the critical nature of the data being stored, do not use RAID level 0 as the RAID level of a mirror repository volume.

Creating an Asynchronous Mirrored Pair

Before you create any mirror relationships, you must create an asynchronous mirror group. The asynchronous mirror group is a logical entity that spans a local storage array and a remote storage array that is used for mirroring and that contains one or more mirrored pairs. The mirrored pairs consist of two volumes: a primary volume on the local storage array and a secondary volume on the remote storage array. If neither the primary volume nor the secondary volume exist, you must create these volumes.

Keep these guidelines in mind when you create an asynchronous mirrored pair:

- The Asynchronous Mirroring premium feature must be enabled and activated on the local and remote storage arrays that you want to used for mirroring.
- The local and remote storage arrays must be connected through a proper Fibre Channel fabric or iSCSI interface.
- The remote storage array must contain a volume that has a capacity that is greater than or equal to the capacity of the volume that is to be used as the primary volume on the local storage array.
- The RAID level of the secondary volume does not have to be the same as the primary volume.
- Make sure you know the passwords for both the local and remote storage arrays.

Use these steps to create an asynchronous mirrored pair.

1. Enable the Asynchronous Mirroring premium feature.
2. Activate the Asynchronous Mirroring premium feature.
3. Create the asynchronous mirror group.

Enabling the Asynchronous Mirroring Premium Feature

The first step in creating an asynchronous mirrored pair is to make sure that the Asynchronous Mirroring premium feature is enabled on both storage arrays. Because Asynchronous Mirroring is a premium feature, you need a feature key file to enable the premium feature. The command for enabling the feature key file is as follows:

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

In this command, the *filename* is the complete file path and file name of a valid feature key file. Enclose the file path and the file name in double quotation marks (“ ”). Valid file names for feature key files end with a `.key` extension.

You can use the **`show storageArry features`** command to list the premium features installed on the storage array.

Activating the Asynchronous Mirroring Premium Feature

Activating the Asynchronous Mirroring premium feature prepares the storage arrays to create and configure mirror relationships. After you activate the premium feature, the secondary ports for each controller are reserved and dedicated to remote mirror use. Any host-initiated I/O operation is not accepted by the dedicated port, and any request received on the dedicated port is accepted only from another controller participating in the mirror relationship.

To activate the Asynchronous Mirroring premium feature, use this command:

The storage array performs the following actions when you activate the Asynchronous Mirroring premium feature:

- Logs out all hosts currently using the highest numbered Fibre Channel host port on the controllers.
- Reserves the highest numbered Fibre Channel host port on the controllers for mirror data transmissions.
- Rejects all host communication to this controller host port as long as the Asynchronous Mirroring premium feature is active.

After you activate the Asynchronous Mirroring premium feature, you must set up an asynchronous mirror group and an asynchronous mirrored pair.

Creating the Asynchronous Mirroring Group

An asynchronous mirror group contains several mirrored pairs so that they can be managed as a single entity. You create an asynchronous mirror group to define the synchronization settings for all mirrored pairs within the group. Each mirrored pair in an asynchronous mirror group shares the same synchronization settings, primary role, secondary role, and write mode.

The asynchronous mirror group is associated with the local storage array and remote storage array that is used for mirroring. The local storage array is the primary side of the asynchronous mirror group, while the remote storage array is the secondary side of the asynchronous mirror group. All volumes added to the asynchronous mirror group on the local storage array hold the primary role in the mirror relationship. Subsequently, all volumes added to the asynchronous mirror group on the remote storage array hold the secondary role in the mirror relationship.

Keep these guidelines in mind when creating an asynchronous mirror group:

- A storage array has a maximum number of asynchronous mirror groups. The maximum number of asynchronous mirror groups depends on your configuration
- Asynchronous mirror groups are created empty and asynchronous mirrored pairs are added to the groups later. You can add only asynchronous mirrored pairs to an asynchronous mirror group. Each mirrored pair is associated with exactly one asynchronous mirror group.
- All storage arrays with the Asynchronous Mirroring premium feature activated are listed and can be used for mirror activities.
- Storage arrays are displayed by their storage array name. If a storage array does not have a name, the storage array is displayed as "Unnamed."

Make sure that you run the **`Create Asynchronous Mirror Group`** command on the local storage array. Asynchronous mirror group creation is initiated from the storage array that contains the volumes that hold the primary role in the mirror relationship. You use the **`Create Asynchronous Mirror Group`** command to specify the remote storage array that contains the volumes that will provide the secondary role in the mirror relationship.

The command has this form:

```
create asyncMirrorGroup userLabel="asyncMirrorGroupName"
```

```
(remoteStorageArrayName="storageArrayName" | remoteStorageArrayWwn="wwID")
interfaceType=(FC | iSCSI)
[remotePassword="password"
syncInterval=integer (minutes | hours | days)
warningSyncThreshold=integer (minutes | hours | days)
warningRecoveryThreshold=integer (minutes | hours | days)
warningThresholdPercent=percentValue
autoResync=(TRUE | FALSE)]
```

When you run this command you create a new, empty asynchronous mirror group on both the local storage array and the remote storage array.

This example shows how to create an asynchronous mirror group on a Windows command prompt:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “create asyncMirrorGroup userLabel="\EngDevData\"
remoteStorageArrayName="\Eng_Backup\"
interfaceType=iSCSI
remotePassword="\xxxxx\"
syncInterval=8 hours
warningSyncThreshold=1 hours
warningRecoveryThreshold=2 hours
warningThresholdPercent=80
autoResync=TRUE]
```

The command in this example creates a new asynchronous mirror group with a repository volume on a remote storage array that already has the name "Eng_Backup." The interface between the local storage array and the remote storage array is iSCSI. The remote storage array is password protected and the password must be used to create the asynchronous mirror group. The synchronization between the local storage array and the remote storage array automatically takes place once every eight hours. If the synchronization cannot be completed successfully the administrator receives a message one hour after the synchronization did not work. When the repository volume has reached 80 percent of capacity, a warning alert is sent to the administrator.

This example shows how to use the command in a script file:

```
create asyncMirrorGroup userLabel="EngDevData"
remoteStorageArrayName="Eng_Backup"
interfaceType=iSCSI
remotePassword="xxxxx"
syncInterval=8 hours
warningSyncThreshold=1 hours
warningRecoveryThreshold=2 hours
warningThresholdPercent=80
autoResync=TRUE]
```

After you create the asynchronous mirror group, you can create the asynchronous mirrored pair to start performing remote mirroring operations.

Creating the Asynchronous Mirroring Pair

After you have created the asynchronous mirror group you can create the mirrored pairs that you want in the asynchronous mirror group. Creating a mirrored pair involves these steps:

1. Determining if you have a candidate secondary mirror volume on the remote storage array.

2. Adding a primary volume that you want to mirror to the asynchronous mirror group on the local storage array.
3. Enabling the relationship between the primary volume on the local storage array and the repository volume on the remote storage array.

When you add a member volume to the primary asynchronous mirror group on the local storage array, the firmware creates a place holder object in the secondary asynchronous mirror group on the remote storage array. The place holder object represents the associated mirror volume. When you add a member volume to the secondary asynchronous mirror group on the remote storage array, the member volume replaces the placeholder object and resolves the mirror configuration.

After you have created the asynchronous mirrored pair you will actually have three volumes in the relationship:

- Primary volume on the local storage array that holds the original data
- Secondary volume on the remote storage array that holds the duplicate data
- Repository volume on the local storage array that holds the data that was changed on the primary volume

If any member volumes of the asynchronous mirror group are actively synchronizing data from a periodic synchronization interval when the new member volume is added, the synchronization process is halted. Existing recovery points from a past completed synchronization process, if any, are preserved. After the new asynchronous mirror group mirror-pair has been initialized, a new consistent set of recovery points are taken of all member volumes of the asynchronous mirror group.

Keep these guidelines in mind when creating the asynchronous mirroring pairs:

- Primary volumes and secondary mirror repository volumes do not need to be the same size.
- Mirror repository volumes are independent of the associated primary volume and secondary volume so that they can be created in separate volume groups with different RAID levels.
- Mirror repository volumes must have the same Data Assurance (DA) and Quality of Service (QoS) settings as the associated volumes. For example, if a mirrored pair is DA enabled, the associated mirror repository volume must be DA enabled.
- Member volumes on both the local storage array and the remote storage array must be standard RAID volumes. They cannot be snapshot volumes, views, or repository volumes.

Determining if You Have a Secondary Mirror Volume

To determine if you have a candidate secondary mirror volume that is appropriate to your needs on the remote storage array, run this command on the remote storage array:

```
show allVolumes summary
```

This command returns information about the number of volumes on the storage array, the name of the volumes, the capacity, RAID level, and other information. If you run the command without the summary parameter, you receive several pages of detailed information. However, running with the summary parameter provides you with enough information to determine what are the volumes that you might be able to use for a repository volume.

Adding a Primary Volume to the Asynchronous Mirror Group

NOTE You cannot add a primary volume to the asynchronous mirror group if the asynchronous mirror group contains orphaned mirror-pair members.

A mirrored relationship between the primary volume on the local storage array and the secondary volume on the remote storage array is established by first adding a member volume to the asynchronous mirror group on the local storage array. To add a member volume to the asynchronous mirror group on the local storage array run this command on the local storage array:

```
add volume="volumeName" asyncMirrorGroup="asyncMirrorGroupName"
remotePassword="password"
(repositoryVolume="repos_xxxx" |
repositoryVolume=(volumeGroupName [capacity=capacityValue])
repositoryVolume=(diskPoolName [capacity=capacityValue]))
```

When you run this command, at a minimum you must perform these actions:

- Identify the volume on the local storage array that you want to mirror to a repository volume on the remote storage array.
- Identify the asynchronous mirror group in which you want to place the volume that you want to mirror.
- Identify an existing repository volume or create a new repository volume.

If an unused repository volume already exists on the remote storage array, you can reuse that repository volume. Otherwise you must create the repository volume. With this command you have the option of creating a repository volume in either a volume group or a disk pool. You identify in which volume group or disk pool you want to place the repository volume and the size of the repository volume. The storage management software and firmware then creates a repository volume with a name "repos_xxxx" where "xxxx" is a numerical identifier. After the repository volume is created, you cannot rename it.

Establishing the Link Between the Primary Volume and the Secondary volume

After you have identified a qualified secondary volume on the remote storage array and added the primary volume on the local storage array, you must link the two volumes. Use this command to link the two volumes:

```
establish asyncMirror volume="secondaryVolumeName"
asyncMirrorGroup="asyncMirrorGroupName"
primaryVolume="primayVolumeName"
```

- `secondaryVolumeName` is the member volume on the remote storage array.
- `asyncMirrorGroupNameName` is the member asynchronous mirror group that contains the mirror-pair.
- `primaryVolumeName` is the member volume on the local storage array.

After you run this command, the asynchronous mirrored pair is linked and the initial mirror is started. During the initial synchronization, all of the data is copied from the primary volume to the secondary volume. During the copy operation, the primary volume is accessible by all hosts for write operations, but the secondary volume is not ready to use for recovery. During the initial synchronization, the repository volume delta log tracks write requests to the primary volume. At the conclusion of the initial synchronization the repository has any changed data and can be used for synchronization between the primary volume and the secondary volume. Because the initial synchronization is a complete copy between the primary volume and the secondary volume, the initial synchronization can take a long time depending on the size of the primary volume. Asynchronous mirroring operations can now be performed as you defined when you created the asynchronous mirror group.

If you want to make any changes to the synchronization settings between the local, primary volume and the remote secondary volume, use the **`set asyncMirrorGroup`** command.

Changing Asynchronous Mirroring Settings

The **`set asyncMirrorGroup`** command enables you to change the property settings for an asynchronous mirrored pair. Use this command to change these property settings:

- Synchronization interval – The length of time between automatically sending updates of modified data from the local storage array to the remote storage array.

- Synchronization warning threshold – The length of time to wait until a warning is triggered when the synchronization of volumes takes longer than the defined time.
- Recovery warning threshold – The length of time to wait until a warning is triggered when the automatic data update for the point-in-time image on the remote storage array is older than a defined time.
- Warning threshold percent – The percent of capacity of the mirror repository volume at which a warning is sent if the mirror repository volume is nearing full.
- User label – A new name for the asynchronous mirror group.
- Automatic resynchronization – A setting to enable or disable automatic resynchronization between the primary volume and the secondary volume.
- Volume – A repository volume for which you want to increase the capacity.
- Repository volume – An unused repository volume that you want to add to another repository volume to increase capacity.
- Role – A setting to change the role, primary or secondary, of the volumes in an asynchronous mirror group.
- Force – The setting to force a role change on the local storage array if the link between the local storage array and the remote storage array is not available.
- No synchronization – A setting to force synchronization before a role change.

This example shows how to use the **`set asyncMirrorGroup`** command to increase the capacity of a repository volume:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “set asyncMirrorGroup \"amg_001\"
volume="\repos_006\" increaseRepositoryCapacity
repositoryVolumes=("\repos_0020\" "\repos_0021\");”
```

This example shows how to use the command in a script file:

```
set asyncMirrorGroup "amg_001"
volume="repos_006" increaseRepositoryCapacity
repositoryVolumes=("repos_0020" "repos_0021");
```

Suspending and Resuming the Asynchronous Mirror Group

Use the **`suspend asyncMirrorGroup`** command to stop data transfer between all of the primary volumes and all of the secondary volumes in an asynchronous mirror group without disabling the asynchronous mirroring relationships. Suspending the asynchronous mirroring relationship lets you control when the data on the primary volume and data on the secondary volume are synchronized. Suspending the asynchronous mirroring helps to reduce any performance impact to the host application that might occur while any changed data on the primary volume is copied to the secondary volume. Suspending the asynchronous mirroring is particularly useful when you want to run a backup of the data on the secondary volume.

When the asynchronous mirroring 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 asynchronous mirroring repository volumes. After the asynchronous mirroring 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.

This example shows the **`suspend asyncMirrorGroup`** command:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “suspend asyncMirrorGroup ["amg_001"];”
```

The asynchronous mirror group name is `amg_001`. You must enclose the name in square brackets. In this example the double quotation marks are optional. The double quotation marks must be used if the asynchronous mirror group name has special characters, such as a colon as used in this example `"amg:001"`. The double quotation marks also must be used if the asynchronous mirror group name is only a number, as in this example `"001"`. Using double quotation marks when not needed does not prevent the command from running successfully. If you have any doubts about the asynchronous mirror group name, use the double quotation marks inside the square brackets.

This example shows how to use the command in a script file:

```
suspend asyncMirrorGroup ["amg_001"];
```

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

This example shows the **`resume asyncMirrorGroup`** command:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “resume asyncMirrorGroup ["amg_001"];”
```

This example shows how to use the command in a script file:

```
resume asyncMirrorGroup ["amg_001"];
```

Manually Resynchronizing Volumes in an Asynchronous Mirror Group

Manually resynchronizing the volumes in an asynchronous mirror group immediately resynchronizes all of the mirror relationships within the asynchronous mirror group. You cannot perform this operation if one of these conditions exists:

- The asynchronous mirror group has failed because any dependent component of the mirror configuration is in a Failed state.
- The asynchronous mirror group is in a Suspended state.

The manual resynchronization request must be issued to the primary asynchronous mirror group.

You can run a manual resynchronization on an asynchronous mirror group that is configured with a periodic synchronization interval. A manual resynchronization does not affect the periodic resynchronization schedule. If you manually resynchronize the asynchronous mirror group when it is performing an initial synchronization, the manual resynchronization command is rejected.

If you run a manual resynchronization when an asynchronous mirror group is in the midst of a periodic synchronization operation, the current process is stopped and the point in time synchronization points on the primary are discarded. The delta log tracking new host writes and the delta log being used for the synchronization process are merged. New point in time synchronization points are created and a new resynchronization is started.

To manually run a resynchronization, run this command:

```
start asyncMirrorGroup synchronize
```

Changing Asynchronous Mirroring Roles

You can change the mirroring roles of an asynchronous mirror group, promoting the secondary storage array to the primary storage array. Each storage array changes roles. First the original primary becomes the secondary, and then the original secondary becomes the primary. As soon as the role change completes, all asynchronous mirror group member volumes on the new-primary storage array become fully accessible for host I/O operations. Follow these guidelines when changing roles:

- You can run the command to change roles from either primary storage array or secondary storage array.
- All of the mirror pairs of the asynchronous mirror group must have a valid recovery point before performing a role reversal.
- The asynchronous mirror group must be in a non-failed state and must not already be in role changing operation.
- You cannot change roles if the asynchronous mirror group contains incomplete or orphaned member volumes.

When the communication between the asynchronous mirroring storage arrays is operating normally, role reversal is coordinated between the local and remote storage arrays. During role reversal these two events occur:

- The original primary volumes are protected from new write requests just as if they were secondary volumes.
- A resynchronization process from the original primary volumes to the original secondary volumes starts.

The resynchronization operation completes after all mirror-pairs of the asynchronous mirror group are fully synchronized. If a scheduled synchronization operation is in progress when a controller receives the role change request, the resynchronization operation is stopped and restarted so that all regions flagged in both the host-write tracking delta log and the synchronization delta log are copied to the secondary volumes.

After the synchronization finishes and the roles are changed these actions take place:

- Recovery points are created on the original primary storage array that has now become the secondary storage array.
- The volumes on the new primary storage array change state to Optimal and operate normally: servicing read and write requests, tracking write requests, and periodically synchronizing to the secondary volumes.

To change roles, use this form of the **`set asyncMirrorGroup`** command.

```
set asyncMirrorGroup ["asyncMirrorGroupName"] role=(primary | secondary)
```

The `role` parameter enables you to define the role you want from either the primary storage array or the secondary storage array. For example, if you are on the primary storage array and you want to change roles, use the `secondary` parameter. The command would look like this:

```
set asyncMirrorGroup ["asyncMirrorGroupName"] role=secondary
```

As soon as you run this command, the storage arrays start changing roles.

Two optional parameters support changing the roles of the primary storage array and the secondary storage array:

- `force`
- `nosync`

The `force` parameter forces a role change if the communications link between the storage arrays is down and promotion or demotion on the local side results in a dual-primary condition or a dual-secondary condition. To force a role reversal, set this parameter to **TRUE**. The default value is **FALSE**.

The `nosync` parameter defines whether to perform an initial synchronization between the primary storage array and the secondary storage array before a role reversal operation is performed. To perform an initial synchronization, set this parameter to **TRUE**. The default value is **FALSE**.

If the role change is interrupted because of a communication failure between the storage arrays, the mirror roles can possibly end as two secondary roles. This role conflict does not compromise the data synchronization state.

Canceling a Pending Asynchronous Mirror Group Role Change

You can cancel a pending role change by running this command:

```
stop asyncMirrorGroup rolechange
```

This command restores the asynchronous mirror group to the normal operating state including write requests to the volumes on the primary volume. The synchronization process that was started as part of the role change operation is allowed to finish. The next periodic resynchronization is scheduled based on the most recently completed periodic synchronization and the current asynchronous mirror group synchronization settings.

Resolving Role Conflicts

Because you can force a change in the role of the storage arrays in an asynchronous mirror group, sometimes this might result in a condition where you have two primary or two secondary roles at the same time. This usually occurs when communications between the storage arrays cannot take place. For example, the original primary side might be operational, but cannot be reached because of a link failure. In this case, a forced promotion of the secondary to become a primary results in both sides being able to receive new data write requests since the most recent resynchronization. Later, the original primary site might be reactivated or connectivity reestablished resulting in both local and remote storage arrays viewing themselves as the primary.

If you have a role conflict, your asynchronous mirror group might have two primary volumes or two secondary volumes in an asynchronous mirror group. If you have two primary volumes, both volumes can accept host writes, but you do not have any mirror volumes. If you have two secondary volumes, neither volume can accept host writes. In either case you no longer have a valid asynchronous mirror group.

If such a role conflict occurs, you need to transition the mirrored pair back to a state in which one side is clearly recognized as primary and the other as secondary. To transition back to a valid asynchronous mirror group, use this command.

```
set asyncMirrorGroup ["asyncMirrorGroupName"] role=(primary | secondary)
```

Run this command on one of the storage arrays in the asynchronous mirror group and define the role of that storage array as you need to meet your mirroring requirements. Define the storage array as either the storage array with the primary volumes or the storage array with the secondary volumes.

Removing Volumes from the Asynchronous Mirror Group

When you have an asynchronous mirror group you have three volumes that you need to manage:

- Primary on the local storage array

- Secondary on the remote storage array
- Repository on both storage arrays

As part of management actions, you might want to remove a volume from an asynchronous mirror group. Use the **`remove volume asyncMirrorGroup`** command to remove the link between a primary volume and a secondary volume.

Removing a volume from an asynchronous mirror group disassociates the volume from the asynchronous mirror group and also disassociates the corresponding volume on the remote storage array from the asynchronous mirror group. Removing a volume disassociates the mirror repositories from the affected volumes on both of the local and remote storage arrays. You can then either delete the repository volume or you can keep the repository volume for later use as a repository volume for a different configuration. If you choose to keep the repository volume, you can use it as the principal repository volume for another asynchronous mirrored pair or to increase the capacity of another repository volume.

The volume on the primary asynchronous mirror group is removed first to halt any in-progress synchronization I/O operations. If the remote storage array is not accessible because of an inter-storage array communication problem, you can force the removal operation so that only the volume on the local asynchronous mirror group is removed. The corresponding volume on the remote storage array remains in the remote asynchronous mirror group.

Removing the link between a primary volume and a 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. Later, you can establish the mirror relationship between the two volumes and resume normal mirror operations. You can remove the mirror relationship for one or several remote mirrored pairs with this command.

This example shows the **`remove volume asyncMirrorGroup`** command:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “remove volume [\"Jan_04_Account\"] asyncMirrorGroup=\"amg_001\"
deleteRepositoryMembers=TRUE;”
```

The command in this example removes a volume with the name `Jan_04_Account` from the asynchronous mirror group with the name amg_001. Because the `deleteRepositoryMembers` is set to TRUE, the repository volume is deleted. You must enclose the volume name in double quotation marks inside square brackets. Also, you must enclose the asynchronous mirror group name in double quotation marks.

This example shows how to use the command in a script file:

```
remove volume ["Jan_04_Account"] asyncMirrorGroup="amg_001"
deleteRepositoryMembers=TRUE;
```

You must run this command on the local storage array to remove the primary volume.

If the volume is not successfully removed from both sides of the asynchronous mirror group, the mirror volume that was not removed becomes an orphan. Orphans are detected when communications between the controller on the local storage array and the corresponding controller on the remote storage array is restored. At that time the two sides of the mirror configuration reconcile mirror parameters. Some synchronization operations function normally when a asynchronous mirror group contains orphaned members, but some configuration operations, such as role change, cannot be performed when the asynchronous mirror group contains orphaned members. To avoid this situation, remove the incomplete mirror volumes by running this command:

```
remove asyncMirrorGroup ["asyncMirrorGroupName"]
incompleteMirror volume="volumeName"
```

Deleting an Asynchronous Mirror Group

If you need to remove an asynchronous mirror group for any reason, you can remove it using this command:

```
delete asyncMirrorGroup
```

Before you can delete an asynchronous mirror group you must remove the primary volume, the secondary volume, and the repository volume from the asynchronous mirror group. The asynchronous mirror group must be completely empty before you can delete it. You can remove these volumes using this command:

```
remove volume asyncMirrorGroup
```

Removing these volumes from the asynchronous mirror group does not delete the volumes. The data is not lost.

Deleting the asynchronous mirror group first removes the remote asynchronous mirror group then removes the local asynchronous mirror group. You can run the **`delete asyncMirrorGroup`** command from either the primary asynchronous mirror group or secondary asynchronous mirror group.

Chapter 8 - Using the Synchronous Mirroring Feature

The Synchronous 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 on one storage array, you can promote the second storage array to take over responsibility for computing services. Synchronous Mirroring is designed for extended storage environments in which the storage arrays that are used for Synchronous 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 you set up the remote mirrored pair. The data transfers occur at Fibre Channel speeds to maintain data on the different storage arrays. Because Synchronous Mirroring is storage based, it does not require any server overhead or application overhead.

You can use Synchronous Mirroring for these functions:

- **Disaster recovery** – Synchronous Mirroring lets you replicate data from one site to another site, which provides an exact duplicate at the remote (secondary) site. If the primary site fails, you can use mirrored data at the remote site for failover and recovery. You can then shift storage operations to the remote site for continued operation of all of the services that are usually provided by the primary site.
- **Data vaulting and data availability** – Synchronous Mirroring lets you 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** – Synchronous Mirroring provides the ability to have two storage arrays back up each other by duplicating critical volumes on each storage array to volumes on the other storage array. This action lets each storage array recover data from the other storage array in the event of any service interruptions.

How Synchronous Mirroring Works

When you create a remote-mirror pair, the remote-mirror 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 might only be included in one mirrored volume pair.

Table 25. Maximum Number of Defined Mirrors per Storage Array

Base System	Max # Defined Mirrors
QS1200	128
QS2400	128
QD6000	128
QD7000	128 (synchronous mirror only, no legacy asynchronous mirror)

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 controller owner of the primary volume but will not accept host write requests. Hosts are able to read from the secondary volume, which appears as read-only.

In the event of a disaster or a catastrophic failure at the primary site, you can perform a role reversal to promote the secondary volume to a primary role. Hosts then are able to read from 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 that is created as a resource for the controller owner of the primary volume in a remote mirrored pair. The controller stores mirror 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 the accidental powering down of the storage arrays.

When you activate the Synchronous 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 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, do not use RAID Level 0 as the RAID level of mirror repository volumes. The required size of each volume is 128 MB, or 256 MB total for both mirror repository volumes of a dual-controller storage array. In previous versions of the Synchronous Mirroring premium feature, the mirror repository volumes required less disk storage space and needed to be upgraded to use the maximum amount of mirror relationships.

Mirror Relationships

Before you create a mirror relationship, you must enable the Synchronous Mirroring premium feature on both the primary storage array and the secondary storage array. 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 that contains 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

The controllers manage data replication between the primary volume and the secondary volume. This process is transparent to host machines and applications. This section describes how data is replicated between the storage

arrays that are participating in Synchronous Mirroring. This section also describes 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 Synchronous Mirroring premium feature provides two write mode options that affect when the I/O completion indication is sent back to the host: Synchronous and Asynchronous.

Synchronous Write Mode

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 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. The controller then sends an I/O completion indication back to the host system. Synchronous write mode is selected as the default value and is the recommended write mode.

Asynchronous Write Mode

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 might become inconsistent with the data on the primary volume. This event could jeopardize any attempt to recover data if a disaster occurs on the primary storage array.

Write Consistency Mode

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 that is in the Write consistency mode. The data on the secondary, remote storage array cannot be considered fully synchronized until all of the remote mirrors that are in the Write consistency mode are synchronized.

If one mirror relationship in the group becomes unsynchronized, all of the mirror relationships in the group become unsynchronized. Any write activity to the remote, secondary storage arrays is prevented to protect the consistency of the remote data set.

Link Interruptions or Secondary Volume Errors

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

In this case, the remote write operation cannot be completed to the secondary volume, and the primary volume and the secondary volume 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 do 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 takes place. This resynchronization happens automatically, or it must 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 starts, the mirrored pair transitions from an Unsynchronized status to a Synchronization in Progress status.

The primary controller also marks the mirrored pair as unsynchronized when a volume error on the secondary side prevents the remote write from completing. For example, an offline secondary volume or a failed secondary volume can cause the remote 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. The mirrored pair then transitions to a Synchronization in Progress status.

Resynchronization

Data replication between the primary volume and the 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. The controller 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 a volume error prevents communication with the secondary storage array, the controller owner of the primary volume transitions the mirrored pair into an Unsynchronized status. The controller owner then 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 do 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 that changed during the interruption to the secondary volume. Only the blocks of data that have changed on the primary volume during the link interruption are copied to the secondary volume.

ATTENTION Possible loss of data access – Any communication disruptions between the primary storage array and the secondary storage array while resynchronization is underway could result in a mix of new data and old data on the secondary volume. This condition would render the data unusable in a disaster recovery situation.

Creating a Synchronous Mirroring Pair

Before you create any mirror relationships, volumes must exist at both the primary site and the secondary site. The volume that resides on the local storage array is the primary volume. Similarly, the volume that resides on the remote storage array is the secondary volume. If neither the primary volume nor the secondary volume exist, you must create these volumes. Keep these guidelines in mind when you create the secondary volume:

- 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.

Use these steps to create the volume.

1. Enable the Synchronous Mirroring premium feature.

2. Activate the Synchronous Mirroring premium feature.
3. Determine candidates for a remote mirrored pair.
4. Create the remote mirrored pair.

Performance Considerations

Keep these performance considerations in mind when you create 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, full synchronization 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. To set the synchronization priority level, consider these guidelines:
  - A full synchronization at the lowest synchronization priority level takes approximately eight times as long as a full synchronization at the highest synchronization priority level.
  - A full synchronization at the low synchronization priority level takes approximately six times as long as a full synchronization at the highest synchronization priority level.
  - A full synchronization at the medium synchronization priority level takes 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 takes 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 mirror relationship.
- The time that it takes for data to be copied from the primary volume to the secondary volume might impact overall performance. This impact is primarily caused by the delay and system resource required for copying data to the remote mirror. Some delay might also occur because of the limit to the number of simultaneous writes.

Activating the Synchronous Mirroring Feature

Activating the Synchronous Mirroring 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 mirror use. In addition, a mirror repository volume is automatically created for each controller in the storage array. As part of the activation process, you can 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 mirror repository volumes.

The free capacity that you select for the mirror repository volume must have a total of 256 MB of 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 returns an error message that gives the amount of space needed for the mirror repository volumes. The command does not try to activate the Synchronous Mirroring premium feature. You can re-enter the command using the value from the error message for the repository storage space value.

The RAID level that you choose for the mirror repository volume has these constraints:

- **RAID Level 0** – You cannot use RAID level 0.
- **RAID Level 1** – The number of drives must be an even number. If you select an odd number of drives, the controller firmware returns an error.
- **RAID Level 5** – You must have a minimum of three drives in the volume group.
- **RAID Level 6** – You must have a minimum of five drives in the volume group.

To activate the Synchronous Mirroring feature, use this command:

```
activate storageArray feature=syncMirror
```

The **`activate storageArray feature=syncMirror`** command provides three methods for defining the drives for your mirror repository volume:

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

Activating the Synchronous Mirroring Feature with User-Assigned Drives

Activating the Synchronous Mirroring feature by assigning the drives provides flexibility in defining your configuration by letting you choose from the available drives in your storage array. Choosing the drives for your remote mirror automatically creates a new volume group. You can specify which drives to use and the RAID level for the new volume group.

The command takes this form:

This example shows a command in which you assign the drives:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “activate storageArray feature=syncMirror
repositoryRAIDLevel=5
repositoryDrives=(1,1 1,2 1,3 1,4 1,5);”
```

The command in this example creates a new mirror repository volume consisting of five drives that form a new volume group. The new volume group has RAID Level 5.

This example shows how to use the command in a script file:

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

Activating the Synchronous Mirroring Feature with Software-Assigned Drives

With this version of the **`activate storageArray feature=syncMirror`** command, you choose an existing volume group in which to place the mirror repository volume. The storage management software then determines which drives 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 command takes this form:

```
activate storageArray feature=syncMirror
```

```
repositoryVolumeGroup=volumeGroupNumber
[freeCapacityArea=freeCapacityIndexNumber]
```

This example shows a command in which the software assigns the drives:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “activate storageArray feature=syncMirror
repositoryVolumeGroup=2 freeCapacityArea=2;”
```

The command in this example creates a new mirror repository volume in volume group 2 using the second free capacity area.

This example shows how to use the command in a script file:

```
activate storageArray feature=syncMirror
repositoryVolumeGroup=2 freeCapacityArea=2;
```

Activating the Synchronous Mirroring Feature by Specifying a Number of Drives

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

This example shows a command in which you specify the number of drives:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “activate storageArray feature=syncMirror
repositoryRAIDLevel=5 repositoryDriveCount=5
driveType=SAS;”
```

The command in this example creates a new mirror repository volume by using five software-selected drives for the mirror repository volume. The mirror repository volume has RAID Level 5. The type of drive for the mirror repository volume is SAS.

This example shows how to use the command in a script file:

```
activate storageArray feature=syncMirror
repositoryRAIDLevel=5 repositoryCount=5
driveType=SAS;
```

Determining Candidates for a Remote Mirrored Pair

All of the volumes and drives on the remote storage array might not be available for use as secondary volumes. To determine which volumes on a remote storage array that you can use as candidates for secondary volumes, use the **`show remoteMirror candidates`** command. This command returns a list of the volumes that you can use when creating a remote mirror.

The command takes this form:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “show remoteMirror candidates primary=\“volumeName\”
remoteStorageArrayName=\“storageArrayName\”;”
```

In this command, *volumeName* is the name of the volume that 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 quotation marks (“ ”).

Creating a Remote Mirrored Pair

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

The command takes this form:

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

NOTE You can use the optional parameters as needed to help define your configuration.

This example shows the **`create remoteMirror`** command:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-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 mirror in which the primary volume is named Jan_04_Account on the local storage array. 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 mirrored 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 mirrored 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. These actions help to make sure that the data on the secondary volume is as accurate a copy of the data on the primary volume as possible. The highest synchronization priority does, however, use more system resources, which can reduce system performance.

This example shows how to use the command in a script file:

```
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 mirror, you can see the progress of data synchronization between the primary volume and the secondary volume by running the **`show remoteMirror synchronizationProgress`** command. This command shows the progress as a percentage of data synchronization that has completed.

Changing Synchronous Mirroring Settings

The **set remoteMirror** command lets you change the property settings for a remote mirrored pair. Use this command to change these property settings:

- The volume role (either primary or secondary)
- The synchronization priority
- The write order
- The write mode

You can apply the changes to one or several remote mirrored pairs by using this command. Use the primary volume name to identify the remote mirrored pairs for which you are changing the properties.

This example shows how to use the **set remoteMirror** command:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “set remoteMirror localVolume [Jan_04_Account]
syncPriority=medium
writeOrder=notpreserved
writeMode=asynchronous;”
```

This example shows how to use the command in a script file:

```
set remoteMirror localVolume [Jan_04_Account]
syncPriority=medium
writeOrder=notpreserved
writeMode=asynchronous;
```

Suspending and Resuming a Synchronous Mirroring Relationship

Use the **suspend remoteMirror** command to stop data transfer between a primary volume and a secondary volume in a mirror relationship without disabling the mirror relationship. Suspending a mirror relationship lets you control when the data on the primary volume and data on the secondary volume are synchronized. Suspending a mirror relationship helps to reduce any performance impact to the host application that might 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 backup 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 mirror that is set up in the Write consistency mode, you suspend all remote mirrored pairs within the group. You can then resume mirror operations for any of the individual remote mirrored pairs that are in the group.

This example shows the **suspend remoteMirror** command:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “suspend remoteMirror primary 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**.

This example shows how to use the command in a script file:

```
suspend remoteMirror volume Jan_04_Account
writeConsistency=false;
```

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

This example shows the **`resume remoteMirror`** command:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “resume remoteMirror volume Jan_04_Account
writeConsistency=false;”
```

The `writeConsistency` parameter in this command operates the same as in the previous command.

This example shows how to use the command in a script file:

```
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 a secondary volume. (Removing a mirror relationship is similar to deleting a mirror relationship.) Removing the link between a primary volume and a 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. Later, you can establish the mirror relationship between the two volumes and resume normal mirror operations. You can remove the mirror relationship for one or several remote mirrored pairs with this command.

This example shows the **`remove remoteMirror`** command:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “remove remoteMirror localVolume [Jan_04_Account];”
```

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

This example shows how to use the command in a script file:

```
remove remoteMirror localVolume [Jan_04_Account];
```

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

Deleting a Primary Volume or a Secondary Volume

Use the **`delete volume`** command to remove a primary volume or a 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. You cannot redefine the mirror relationship until you create a new volume or choose an alternate volume to replace the deleted volume.

ATTENTION Possible loss of data access – Deleting a primary volume or a secondary volume permanently removes the data from the storage array.

Deactivating the Synchronous Mirroring Feature

If you no longer require the Synchronous Mirroring feature and you have removed all of the mirror relationships, you can deactivate the feature. Deactivating the feature re-establishes the normal use of dedicated ports on both storage arrays and deletes both mirror repository volumes. To deactivate the Synchronous Mirroring feature, use this command:

```
deactivate storageArray feature=remoteMirror
```

Interaction with Other Features

You can run the Synchronous Mirroring feature while running these features:

- Storage Partitioning
- Snapshot (Legacy)
- Volume Copy

When you run the Synchronous Mirroring feature with other features, you must consider the requirements of the other features to help make sure that you set up a stable storage array configuration.

In addition to running with the features, you can also run the Synchronous Mirroring feature while running Dynamic Volume Expansion (DVE).

Storage Partitioning

Storage Partitioning is a feature that lets hosts share access to volumes in a storage array. You create a storage partition when you define any of these logical components in a storage array:

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

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

When you create storage partitions, define the storage partitions after you have created the primary volume and the secondary volume in a Synchronous Mirroring configuration. The storage partition definitions for the primary storage array and the secondary storage array are independent of each other. If these definitions are put in place while the volume is in a secondary role, the administrative effort associated with the site recovery is reduced if it becomes necessary to promote the volume to a primary role.

Snapshot (Legacy) Volumes

A snapshot (legacy) is a point-in-time image of a volume. Typically, it is created so that an application, such as a backup application, can access the snapshot (legacy) volume and read the data while the base volume stays online and is 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 (legacy) repository volume stores information about all data that changed since the snapshot (legacy) was created.

In this version of the storage management software, you can create snapshot (legacy) volumes based on the primary volume or secondary volume of a remote mirror.

Volume Copy

The Volume Copy 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 perform these functions:

- Copy data from volume groups that use smaller-capacity drives to volume groups that use larger-capacity drives
- Back up data
- Restore snapshot (legacy) volume data to the base volume.

You can use a primary volume in a remote mirror as a source volume or a target volume in a volume copy. You cannot use a secondary volume as a source volume or a target volume.

NOTE If you start a role reversal during a copy-in-progress, the copy fails and cannot be restarted.

Dynamic Volume Expansion

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

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

A DVE operation can be performed on a primary volume or a secondary volume of a mirror relationship.

NOTE Although the storage management software indicates that the volume has increased capacity, its usable capacity is the size of the smaller of the primary volume or the secondary volume.

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

Chapter 9 - Using the Volume Copy Feature

The Volume Copy premium feature lets you copy data from one volume (the source) to another volume (the target) in a single storage array. You can use this premium feature to perform these tasks:

- Back up data
- Copy data from one volume to another volume
- Restore snapshot (legacy) 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 the 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 the target volume. If one controller does not own both the source volume and the 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 is 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 are rejected.

The following are valid source volumes:

- A standard volume
- A snapshot (legacy) volume
- The base volume of a snapshot (legacy) volume
- A primary volume that is participating in a remote-mirror pair

The following are not valid source volumes:

- A secondary volume that is participating in a remote-mirror pair
- A snapshot (legacy) repository volume
- A mirror repository volume
- A failed volume

- A missing volume
- A volume currently in a modification operation
- A volume that is holding a Small Computer System Interface-2 (SCSI-2) reservation or a persistent reservation
- A volume that is a source volume or a target volume in another volume copy that has a status of In Progress, Pending, or Failed

Target Volume

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

ATTENTION Possible loss of data access – A volume copy overwrites data on the target volume. Before you start a new operation, make sure that you no longer need the old data, or you have backed up the old data on the target volume.

While the volume copy has a status of In Progress, Pending, or Failed, the controllers reject read and write requests to the target volume. After the volume copy operation is finished, the target volume automatically becomes read only to the hosts, and write requests to the target volume are rejected. You can change the Read-Only attribute after the volume copy has completed or has been stopped. (For more information about the Read-Only attribute, see Viewing Volume Copy Properties.)

The following volumes are valid target volumes:

- A standard volume
- The base volume of a disabled snapshot (legacy) volume or failed snapshot (legacy) volume
- A primary volume that is participating in a remote-mirror pair

The following volumes are not valid target volumes:

- The base volume of an active snapshot (legacy) volume
- A snapshot (legacy) volume
- A mirror repository volume
- A snapshot (legacy) repository volume
- A secondary volume in a remote-mirror pair
- A failed volume
- A missing volume
- A volume with a status of Degraded
- A volume that is currently in a modification operation
- A volume that is holding a SCSI-2 reservation or a persistent reservation
- A volume that is a source volume or a 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 that hold 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 HBA host ports, which provides 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 operation, the resources of the storage array might be diverted from processing I/O activity to completing a volume copy, which might affect the overall performance of the storage array.

These factors contribute to the performance of the storage array:

- The I/O activity
- The volume RAID level
- The volume configuration (number of drives in the volume groups and cache parameters)
- The volume type (snapshot (legacy) volumes might take more time to copy than standard volumes)

When you create a new volume copy, you define the copy priority to determine how much controller processing time is allocated for a volume copy compared with 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 might be affected. The lowest priority rate supports I/O activity, but the volume copy takes longer. You define the copy priority when you create the volume copy pair. You can redefine the copy priority later by using the **`set volumeCopy`** command. You can also redefine the volume copy priority when you recopy a volume.

Restrictions

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

- While a volume copy operation has a status of In Progress, Pending, or Failed, the source volume is available for read activity only. After the volume copy finishes, read activity from and write activity to the source volume is 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 depends upon the storage array configuration.
- A volume that is reserved by the host cannot be selected as a source volume or as a target volume.
- A volume with a status of Failed cannot be used as a source volume or as a target volume.
- A volume with a status of Degraded cannot be used as a target volume.
- You cannot select a volume that is participating in a modification operation as a source volume or as a target volume. Modification operations include Dynamic Capacity Expansion (DCE), Dynamic RAID Level Migration (DRM), Dynamic Segment Sizing (DSS), Dynamic Volume Expansion (DVE), and defragmenting a volume group.

Volume Copy Commands

The following table lists the Volume Copy commands and briefly describes what the commands do.

Table 26. Volume Copy Commands

Command	Description
`create volumeCopy`	Creates a volume copy and starts the volume copy operation.
`recopy volumeCopy`	Re-initiates a volume copy operation using an existing volume copy pair.
`remove volumeCopy`	Removes a volume copy pair.
`set volumeCopy`	Defines the properties for a volume copy pair.
`show volumeCopy`	Returns information about volume copy operations. You can retrieve information about a specific volume copy pair, or all of the volume copy pairs in the storage array.
`show volumeCopy sourceCandidates`	Returns information about the candidate volumes that you can use as the source for a volume copy operation.
`show volumeCopy targetCandidates`	Returns information about the candidate volumes that you can use as the target for a volume copy operation.
`stop volumeCopy`	Stops a volume copy operation.

Creating a Volume Copy

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

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

To create a volume copy, perform these general steps:

1. Determine the candidates for a volume copy.
2. Create the target volume and the source volume for the volume copy.

Determining Volume Copy Candidates

All volumes and drives might not be available for use in volume copy operations. To determine which candidate volumes on the storage array that you can use as a source volume, use the commands in the following table.

Action	Use This CLI Command
Determine which candidate volumes on the storage array you can use as a source volume	`show volumeCopy sourceCandidates`

Action	Use This CLI Command
Determine which candidate volumes on the storage array you can use as a target volume	**show volumeCopy targetCandidates**

The **show volumeCopy sourceCandidates** command and the **show volumeCopy targetCandidates** command each return a list of the capacity information for the source volume candidates and the target volume candidates, respectively.

Creating a Volume Copy

ATTENTION Possible loss of data access – A volume copy overwrites data on the target volume. Make sure that you no longer need the data or have backed up the data on the target volume before you start a volume copy operation.

When you create a volume copy, you must define which volumes that you want to use for the source volume and the target volume. You define the source volume and the target volume 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 command has this form:

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

Before you run the **create volumeCopy** command, perform these actions:

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

This example shows the **create volumeCopy** command:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c "create volumeCopy source=\"Jaba_Hut\" target=\"Obi_1\"
copyPriority=medium targetrReadOnlyEnabled=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 resources that are 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, making sure that the data on the target volume stays unaltered.

This example shows how to use the command in a script file:

```
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 are rejected, unless you disable the Read-Only attribute by using the **set volumeCopy** command.

To view the progress of a volume copy, use the **`show storageArray longRunningOperations`** command. For a long-running operation that is currently running on a volume, this command returns information about the volume action and the amount of the long-running operation that is 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).

Viewing Volume Copy Properties

Use the **`show volumeCopy`** command to view information about one or more selected source volumes or target volumes. This command returns these values:

- The role
- The copy status
- The start time stamp
- The completion time stamp
- The copy priority
- The Read-Only attribute setting for the target volume
- The source volume World Wide Identifier (WWID) or the target volume WWID

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

The command has this form:

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

This example shows the **`show volumeCopy`** command:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “show volumeCopy source [\”JabaHut\”];”
```

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

This example shows how to use the command in a script file:

```
show volumeCopy source [“Jaba_Hut”];
```

Changing Volume Copy Settings

The **`set volumeCopy`** command lets you change these property settings for a volume copy pair:

- The copy priority
- The 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 might be affected. The lowest priority supports I/O activity, but the volume copy takes longer. You can change the copy priority at these times:

- Before the volume copy operation starts
- While the volume copy operation has a status of In Progress
- After the volume copy operation has completed when re-creating a volume copy operation by 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 to make sure that the copied data on the target volume is not corrupted by additional writes to the target volume after the volume copy is created. You want to maintain the read-only status when you are performing these tasks:

- Using the target volume for backup purposes
- Copying data from one volume group to a larger volume group for greater accessibility
- Planning to use the data on the target volume to copy back to the base volume in case of a disabled snapshot (legacy) volume or failed snapshot (legacy) volume

At other times, you might 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, the read request and the write request to the target volume are rejected while the volume copy operation has a status of In Progress, Pending, or Failed.

The command has this form:

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

NOTE You can use the parameters as needed to help define your configuration.

This example shows the **set volumeCopy** command:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “set volumeCopy target [\”Obi_1\”]
copyPriority=highest
targetReadOnlyEnabled=FALSE;”
```

This example shows how to use the command in a script file:

```
set volumeCopy target [“Obi_1”] copyPriority=highest
targetReadOnlyEnabled=FALSE;
```

Recopying a Volume

Use the **recopy volumeCopy** command to 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. Then, you can copy the backup to tape for off-site storage. When you use the **recopy volumeCopy**

command to make a backup, you cannot write to the source volume while the recopy operation is running. The recopy operation might 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.

ATTENTION Possible loss of data access – The **recopy volumeCopy** command overwrites existing data on the target volume and makes the target volume read-only to hosts. The **recopy volumeCopy** command fails all of the snapshot (legacy) volumes that are associated with the target volume, if any exist.

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

The command has this form:

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

NOTE You can use the optional parameters as needed to help define your configuration.

This example shows the **recopy volumeCopy** command:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “recopy volumeCopy target [\”Obi_1\”] copyPriority=highest;”
```

The command in this example copies data from the source volume that is 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. By 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 has changed since the previous copy was made.

This example shows you how to use the command in a script file:

```
recopy volumeCopy target [“Obi_1”] copyPriority=highest;
```

Stopping a Volume Copy

The **stop volumeCopy** command lets you 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 by using the original volume copy pair. After you stop a volume copy operation, all of the mapped hosts will have write access to the source volume.

The command has this form:

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

This example shows the **stop volumeCopy** command:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “stop volumeCopy target [\”Obi_1\”];”
```

This example shows how to use the command in a script file:

```
stop volumeCopy target [“Obi_1”];
```

Removing Copy Pairs

The **`remove volumeCopy`** command lets you remove a volume copy pair from the storage array configuration. All of the volume copy-related information for the source volume and the target volume is removed from the storage array configuration. The data on the source volume or the 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 command has this form:

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

This example shows the **`remove volumeCopy`** command:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c “remove volumeCopy target [\”Obi_1\”];”
```

This example shows how to use the command in a script file:

```
remove volumeCopy target [“Obi_1”];
```

Interaction with Other Features

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

- Storage Partitioning
- Snapshot (Legacy)
- Synchronous Mirroring

When you are running the Volume Copy feature with other features, you must consider the requirements of other features to help make sure that you set up a stable storage array configuration.

In addition to running the Volume Copy feature with other features, you also can run the Volume Copy feature while running Dynamic Volume Expansion (DVE).

Storage Partitioning

Storage Partitioning is a premium feature that lets hosts share access to volumes in a storage array. You create a storage partition when you define any of these logical components in a storage array:

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

The volume-to-LUN mapping lets you 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 make sure that the data is preserved. Hosts that have been mapped to a target volume do not have write access to the volume, and any attempt to write to the read-only target volume results 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 (Legacy) Volumes

A snapshot (legacy) is a point-in-time image of a volume. It is usually created so that an application, such as a backup application, can access the snapshot (legacy) volume and read the data while the base volume stays online and is 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 (legacy) repository volume stores information about all of the data that changed since the snapshot (legacy) was created.

You can select snapshot (legacy) volumes as the source volumes for a volume copy. This selection is a good use of this premium feature, because it performs complete backups without significant impact to the storage array I/O. Some I/O processing resources are lost to the copy operation.

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

When you create a snapshot (legacy) volume, a snapshot (legacy) repository volume is automatically created. The snapshot (legacy) repository volume stores information about the data that has changed since the snapshot (legacy) volume was created. You cannot select a snapshot (legacy) repository volume as a source volume or a target volume in a volume copy.

You can use the Snapshot (Legacy) Volume premium feature with the Volume Copy premium feature to back up data on the same storage array and to restore the data on the snapshot (legacy) volume back to its original base volume.

Chapter 10 - Using the SSD Cache Feature

NOTE Currently the SSD cache feature is not supported with the Quantum QS1200, QS2400, QD6000, and QD7000 base systems.

The SSD Cache feature provides a way to improve read-only performance. SSD cache is a set of Solid-State Disk (SSD) drives that you logically group together in your storage array to implement a read cache for end-user volumes. SSD cache is a secondary cache for use with the primary cache in the controller DRAM. In controller cache, the data is stored in DRAM after a host read. In SSD cache, the data is copied from user-specified base volumes and stored on two internal RAID volumes (one per controller) that are automatically created when you create an SSD cache. These volumes are used for internal cache processing purposes. These volumes are not accessible or displayed in the user interface. However, these two volumes do count against the total number of volumes allowed in the storage array.

NOTE The SSD cache feature is available only when the SSD Cache premium feature is enabled, an SSD cache entity is created, and each base volume for which you want to use the SSD cache has the SSD cache attributes enabled.

Using high performance SSDs to cache the data from the base volumes improves the application I/O performance and response times, and delivers sustained performance improvement across different workloads, especially for high-IOP workloads. Simple volume I/O mechanisms are used to move data to and from the SSD cache. After data is cached and stored on the SSDs, subsequent reads of that data are performed on the SSD cache, thereby eliminating the need to access the base volume.

These are the characteristics of a workload that benefits from using SSD cache:

- Performance is limited by hard disk drive IOPs.
- There are a high percentage of reads relative to writes.
- A large number of reads that are repeat reads to the same or adjacent areas of the drive.
- The capacity of the data that is repeatedly accessed is smaller than the cache capacity. The more volumes being cached, the higher the probability that the capacity of the data accessed exceeds the capacity of the cache.

The SSD Cache premium feature moves data from the hard drives to SSDs following a host read or write so that a subsequent host read of the same logical block addressing (LBA) can be read directly from SSD with a much lower response time than rereading from hard drives.

When you create an SSD cache, you choose which I/O type (file system, database, or web server) that most closely matches the applications and volumes that will use the SSD cache. In addition, you specify the capacity of the SSD cache from a list of possible candidates consisting of different counts of SSD drives. You also have the option to enable SSD cache on all eligible volumes that are currently mapped to hosts. Lastly, after you create the SSD cache, you can enable or disable it on existing volumes or as part of a new volume creation.

SSD Cache Operations

The following list includes common tasks that you can perform with the SSD cache feature.

- Create
- Performance Modeling
- Locate
- View Associated Physical Components
- Add Drives (Capacity)
- Remove Drives (Capacity)
- Suspend
- Resume

- Delete
- Enabling/Disabling SSD Cache
- Change I/O Type
- Rename

SSD Cache Restrictions

- SSD cache is not supported on snapshots (legacy) volumes and snapshot images.
- The maximum usable SSD cache capacity on a storage array is dependent on the controller's primary cache capacity.
- You cannot remove the last drive in an SSD cache without first deleting the SSD cache.
- Only one SSD cache is supported per storage array.

Creating the SSD Cache, Adding Volumes, and Removing Volumes

Before you create the SSD cache, make sure that suitable SSD drives are available on the storage array. You can achieve the best performance when the working set of the data fits in the SSD cache so that most host reads can be serviced from the lower latency solid state disks instead of the higher latency hard drives (HDDs).

To create the SSD cache run this command:

```
create ssdCache userLabel="ssdCacheName"
drives=(trayID1,drawerID1,
slotID1, ...trayIDn,drawerIDn,slotIDn)
[updateExistingVolumes=(TRUE|FALSE)]
```

To use this command you need only to create a name for the SSD cache and identify the specific SSD drives that you want to include in the cache. After you run the command, all of the volumes in the storage array can use the SSD cache.

If you do not want all of the volumes in the storage array to use the SSD cache, set the `updateExistingVolumes` parameter to **FALSE**. When you create the SSD cache without volumes, you can later assign specific volumes to the SSD cache by running this command for each volume that you want to add:

```
set volume ["volumeName"] ssdCacheEnabled=TRUE
```

You can assign standard volumes, snapshot volumes, or consistency group snapshot volumes. You can assign only one volume at a time to the SSD cache.

If you do not want a specific volume to use the SSD cache, you can remove the volume from the SSD cache by running this command:

```
set volume ["volumeName"] ssdCacheEnabled=FALSE
```

After you have created the SSD cache, you can model the performance of the SSD cache to make sure that the SSD cache is running as required for your storage array.

SSD Cache Performance Modeling

After you create the SSD cache, you can run a performance modeling tool to help determine the best cache size for your storage array.

The performance modeling tool helps you determine the performance improvement for SSD cache capacity when you run a workload on the SSD cache that has the same characteristics as the workload that you run in production. Performance modeling monitors and measures I/O activity for a period of time and estimates performance for various SSD cache sizes. The performance modeling tool then shows the actual performance for the SSD cache that you created.

Depending on the cache capacity and workload, it might take 10 to 20 hours to fully populate the cache. The cache has valid information even after a run of a few minutes, but the longer that you can run the performance modeling tool the more accurate the results of the metric data.

The performance modeling tool provides an estimate of performance using these metrics:

- Cache hit percentage
- Average response time

NOTE Performance modeling does not survive a controller reboot.

Starting and Stopping SSD Cache Performance Modeling

To start a performance modeling operation, use this command:

```
start ssdCache [ssdCacheName] performanceModeling
```

Enclose the identifier in square brackets ([ ]). If the SSD cache name contains special characters or consists only of numbers, you also must enclose the identifier in double quotation marks (" ") inside square brackets.

After you start the performance modeling operation it will continue to run until you stop the operation. To stop the performance modeling operation, use this command:

```
stop ssdCache [ssdCacheName] performanceModeling
```

This command immediately stops collecting the performance modeling data and displays the data that has been collected. The data is displayed in the same window from which you ran the command. If you prefer, you can save the performance modeling data to a file by running this command:

```
stop ssdCache [ssdCacheName] performanceModeling file="fileName"
```

fileName is the file path and the file name to which you want to save the performance modeling data. Enclose the file name in double quotation marks (" "), as shown in this example:

```
file="C:\Program Files\CLI\logs\performance.csv"
```

The default name of the file that contains the performance modeling data is `readLinkStatus.csv`. You can use any file name, but you must use the `.csv` extension.

The performance modeling metric data are not available until you stop the performance modeling operation.

You can see the performance modeling data only after you have stopped the performance modeling operation. You cannot get intermediate results while the modeling is being run. When you stop the performance modeling tool, a graphical representation of the results appears, but you can view the results in tabular format by saving the data to a `.csv` file. To compare different results, you can run several performance modeling operations and save each result to

a `.csv` file for each result. Then you can use a spreadsheet program outside of the storage management software to compare the data from the `.csv` file. The performance modeling tool does not support the loading of saved files.

The .csv File Information

The `.csv` file shows the following information:

- **SSD Cache Capacity (GB)** – The amount of SSD cache capacity.
- **SSD Cache Hits (%)** – Derived from cache hits and total reads.
- **Average Overall Response Time** – This value is calculated by the software and is not the result of averaging the response times for **SSD Reads (Cache Hits)**, **HDD Reads**, and **HDD Writes**.
- **Average Response Time (milliseconds)** – This value is the same as the **Average Overall Response Time** in the next paragraph.
- **SSD Reads (Cache Hits)** – The total number of host reads of SSD cache-enabled volumes that were satisfied from the SSD cache.
- **Average Response Time** – The average response time of the SSD reads (Cache Hits).
- **Percentage of I/Os** – An indication of the percentage of SSD reads (Cache Hits).
- **HDD Reads** – The total number of host reads of SSD cache-enabled volumes.

  Compare the reads relative to writes. The reads need to be greater than the writes for effective SSD cache operation. The greater the ratio of reads to writes the better the operation of the cache.
- **Average Response Time** – The average response time of the HDD reads.
- **Percentage of I/Os** – An indication of the percentage of HDD reads.
- **HDD Writes** – The total number of host writes to SSD cache-enabled volumes.

  Compare the reads relative to writes. The reads need to be greater than the writes for effective SSD cache operation. The greater the ratio of reads to writes the better the operation of the cache.
- **Average Response Time** – The average response time of the HDD Writes.
- **Percentage of I/Os** – An indication of the percentage of HDD Writes.

Understanding SSD Cache Performance Modeling Results

When you run the performance modeling tool with a workload that has the same characteristics as you run in production, the performance modeling tool collects this type of information:

- **Cache hit percentage** – The cache-hit percentage indicates the percentage of all read commands that find data in the SSD cache for each of the cache capacities. For almost all workloads, a cache-hit percentage around 75 percent indicates that you have sufficient capacity. A cache-hit percentage much higher than this means that the workload performance is likely to be improved with more SSD cache capacity. However, for some workloads, a cache-hit percentage around 50 percent or even lower indicates that you have sufficient capacity.
- **Average response time** – The performance modeling tool uses calculated response times from the operation of the SSD cache to estimate the time it takes to run during each external and internal I/O operation. The tool uses these response time measurements along with measurements of I/O operations performed during the run of a workload that is applied to the base volumes with SSD cache enabled. Keep in mind that workload on other volumes can affect these results due to increased response time. The resulting estimates are of the average response time for external I/O operations. When making a decision to increase or decrease the capacity of your current SSD cache, look for an acceptable response time, and then compare that with the response time of your current SSD cache capacity.

You can make changes by using options outside of the performance modeling tool. For example, you can change the I/O characteristic types and enable or disable SSD cache on base volumes. These two parameters control the working set for the performance modeling.You change the I/O characteristic types by selecting these menu options:

Storage > **SSD Cache** > **Change I/O Type**. You enable or disable SSD cache on base volumes by selecting these menu options: **Storage** > **Volume** > **SSD Cache** or by issuing the following command:

```
set ssdCache [ssdCacheName] usageHint=(webServer|dataBase|fileSystem);
```

SSD Cache Management Tasks

As soon as you create the SSD cache all of the volumes assigned to the SSD cache can start using it. The SSD cache is a non-addressable volume that requires little maintenance or management. You can, however, perform these tasks to make sure that the SSD cache runs best to support your storage array:

- View information about the drives, status, and capacity of the SSD cache.
- Locate the drives that physically comprise the SSD cache.
- Add drives to and remove drives from the SSD cache.
- Suspend and resume SSD cache operation.
- Rename the SSD cache.

Viewing Information About the SSD Cache

When you create your SSD cache, you provide a name and select the solid state disk (SSD) drives for the cache. However, the firmware maintains more information about the SSD cache, such as maximum capacity, consumed capacity, size of the drives, and other information. You can see all of the information about the SSD cache by running this command:

```
show ssdCache [ssdCacheName]
```

This command returns information about the SSD cache that is similar to this example.

```
SSD Cache name:  my_cache
Status:                              Optimal
Type:                                Read Only
I/O characteristic type:             File System
Maximum capacity allowed:            1,862.645 GB
Current capacity:                    557.792 GB
Additional capacity allowed          1,304.852 GB
Drive capacities:                    All 278.896 GB
Quality of Service (QoS) Attributes
   Security capable:                 No
   Secure:                           No
   Data Assurance (DA) capable:      No
Associated drives:
Tray     Slot
0        4
0        11
Volumes using SSD cache:             volume_test
```

By reviewing this information, you can make sure that your SSD cache is running as you want it to run. For example, if the current capacity is close to the maximum capacity, you might want to add another drive to your SSD cache.

Adding Drives to and Removing Drives from the SSD Cache

After you have created your SSD cache, you might decide that you do not have enough capacity and want to add a drive. Conversely, you might decide that you have too much capacity and that you want to remove a drive to make the drive available for a volume group or disk pool.

You can increase the capacity of an existing SSD cache by using this command to add solid SSDs:

```
set ssdCache [ssdCacheName]
addDrives=(trayID1,drawerID1,slotID1 ... trayIDn,drawerIDn,slotIDn)
```

You can add one or more SSDs by specifying the location of the drives that you want to add. For high-capacity drive trays, specify the tray ID value, the drawer ID value, and the slot ID value for each SSD that you want to add. For low-capacity drive trays, specify the tray ID value and the slot ID value for each SSD that you want to add. Tray ID values are **0** to **99**. Drawer ID values are **1** to **5**. Slot ID values are **1** to **32**. Enclose the tray ID values, the drawer ID values, and the slot ID values in parentheses.

If you want to remove drives, you can use this command:

```
set ssdCache [ssdCacheName]
removeDrives=(trayID1,drawerID1,slotID1 ... trayIDn,drawerIDn,slotIDn)
```

You cannot remove all of the SSDs from the SSD cache using this command; at least one SSD must remain in the SSD cache. If you want to completely remove the SSD cache, use this command:

```
delete ssdCache [ssdCacheName]
```

When you run this command, all data in the SSD cache is purged.

Locating Drives in the SSD Cache

The drives that comprise the SSD cache might be dispersed throughout the storage array. For maintenance or other reasons, you might be required to find the drives in the SSD cache. You can locate the drives in the SSD cache using this command:

```
start ssdCache [ssdCacheName] locate
```

This command identifies the drives that are logically grouped together to form the SSD cache by blinking the indicator lights on the drives in the SSD cache. To turn off the indicator lights on the drives, use this command:

```
stop ssdCache locate
```

Suspending and Resuming SSD Cache Operation

At times you will need to perform maintenance on the drives in the SSD cache. Such maintenance might include replacing non-optimal drives or upgrading drives. To perform maintenance on the SSD cache, you will need to first suspend the operations of the cache. To suspend operations use this command:

```
suspend ssdCache [ssdCacheName]
```

This command temporarily stops caching for all of the volumes that are using the SSD cache. While caching is stopped, host reads are serviced from the base volumes instead of from the SSD cache.

After performing maintenance, you can restart the SSD cache by using this command:

```
resume ssdCache [ssdCacheName]
```

Renaming the SSD Cache

If you want to change the name of the SSD cache, you can use this command:

`set ssdCache [`*`old_ssdCacheName`*`] userLabel="`*`new_ssdCacheName`*`"`

The old SSD cache name must be enclosed in square brackets. The new SSD cache name, however, must be enclosed in double quotation marks.

Chapter 11 - Maintaining 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. Diagnostic or maintenance capabilities are provided using a variety of commands, particularly those starting with **`set`**.

Routine Maintenance

Routine maintenance involves those tasks that you perform periodically in conjunction with AutoSupport (ASUP) to help make sure that the storage array is running at peak performance or for early detection of conditions that might cause future problems.

Running a Media Scan

NOTE If media scan is disabled at the storage array level, this operation has no effect.

Media scan provides a way of detecting and repairing drive media errors before they are found during a normal read from or write to the drives. Any media scan errors that are detected are reported to the Event Log. The Event Log 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 of the volumes in the storage array that have these conditions:

- Has Optimal status
- Has no modification operations in progress
- Has media scan enabled

Errors that are detected during a scan of a user volume are reported to the Major Event Log (MEL).

ATTENTION If the scan determines that parity is in error, the assumption is that parity is inconsistent and attempts to repair the parity to a consistent state. There is no guarantee that the repair would not make a data inconsistency permanent in parity.

If the controller firmware supports repairs the errors are managed as follows:

- **Redundancy mismatches** – Redundancy errors are also known as data/parity mismatch errors. Redundancy errors are found when both the data on a stripe and the redundancy block can be read, but the redundancy information calculated from the data (the parity) does not match the redundancy information on the media. The redundancy process first retries individual read operations to every drive in the volume to determine if the read response data for the drives has changed. After all of the drives have been read again, the redundancy information is checked for consistency. If the redundancy check is consistent an informational MEL event is logged. If the check finds that inconsistencies still exist, a critical MEL event is logged. The redundancy inconsistency is then addressed based upon the NVSRAM setting: either the parity repair capability is disabled and no further action is taken, or the data is presumed to be correct and the redundancy information is recalculated and written to the appropriate sectors on the drives.
- **Recovered read error** – The drive could not read the requested data on its first attempt. When a recovered read error occurs the data is written, with verification, to the same sectors on the drive. If the write-verify operation to the same sectors fails, the failing sectors are reassigned and the data is written, with verification, to replacement sectors on the drive. If the write-verify operation to the replacement sectors fails, the drive is failed. The error is reported to the Event Log.
- **Unrecovered media error** – The drive could not read the requested data on its first try or on any subsequent retries. The result of this action is that for volumes with redundancy protection, the data is reconstructed, rewritten to the drive, and 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.
- **Unfixable error** – The data could not be read, and parity information or redundancy information could not be used to regenerate it. For example, redundancy information cannot be used to reconstruct data on a degraded volume. The result of this action is that the error is reported to the Event Log.

Two commands defining media scan properties:

- **`set volume`**
- **`set storageArray`**

The **`set volume`** command enables a media scan for a volume. The command has this form:

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

When using this command, the volume names must be enclosed in square brackets. If a volume name also has special characters or numbers, you must enclose the volume name in double quotation marks. When using the WWID, enclose the WWID in double quotation marks (" ") inside angle brackets (< >). Do not include colons in the WWID.

The **`set storageArray`** command defines how frequently a media scan is run on a storage array. The command has this form:

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

The `mediaScanRate` values define the number of days over which the media scan runs. Valid values are **`disabled`**, which turns off the media scan; or **`1`** day to **`30`** days, where **`1`** day is the fastest scan rate, and **`30`** days is the slowest.

A value other than what is shown will not allow the media scan to function. Be aware that a syntax error is not returned if an invalid value is entered, but the command fails.

Running a Redundancy Check

NOTE If media scan is disabled at the storage array level, this operation has no effect.

Redundancy checks are performed when media scans are run if redundancy checking is enabled for a volume. (For a description about how to set up and run media scans, see "Running a Media Scan.") During a redundancy check, all of the data blocks in a volume are scanned, and, depending on the RAID level, deteriorated data is corrected. Correction is performed as follows:

- For RAID level 3, RAID level 5, or RAID level 6 volumes, redundancy is checked and repaired.
- For RAID level 1 volumes, the data is compared between the mirrored drives and data inconsistencies are repaired.
- RAID level 0 volumes have no redundancy.

Before you can run a redundancy check, you must enable redundancy checking by using the **`set volume`** command. The command has this form:

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

When using this command, the volume names must be enclosed in square brackets. If a volume name also has special characters or numbers, you must enclose the volume name in double quotation marks. When using the WWID, enclose the WWID in double quotation marks (" ") inside angle brackets (< >). Do not include colons in the WWID.

Resetting a Controller

IMPORTANT If you are not using a multi-path driver to manage controller access to volumes, resetting the controller means the controller is no longer available for I/O operations until the reset is complete. If a host is using volumes that are owned by the controller being reset, the I/O that is directed to the controller is rejected. Before resetting the controller, either make sure that the volumes that are owned by the controller are not in use, or make sure that a multi-path driver is installed on all of the hosts that are using these volumes.

Resetting a controller is the same as rebooting the controller processors. To reset a controller, use this command:

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

Enabling a Controller Data Transfer

At times, a controller might become quiescent while running diagnostics. If this condition occurs, the controller might become unresponsive. To revive a controller that has become quiescent while running diagnostics, use this command:

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

Removing Persistent Reservations

Persistent reservations preserve volume registrations, and they prevent hosts, other than the host defined for the volume, from accessing the volume. You must remove persistent reservations before you make these 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, use this command:

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

To clear persistent volume reservations, use this command:

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

Synchronizing the Controller Clocks

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

```
set storageArray time
```

Locating Drives

At times, you might need to locate a specific drive. In very large storage array configurations, this task can sometimes be awkward. If you need to locate a specific drive, you can do so by turning on the indicator light on the front of the drive. To locate a drive, use this command:

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

NOTE The `drawerID` parameter is only required for high capacity drive trays.

To turn off the indicator light after locating the drive, use this command:

```
stop drive locate
```

The `drive` parameter supports both high-capacity drive trays and low-capacity drive trays. A high-capacity drive tray has drawers that hold the drives. The drawers slide out of the drive tray to provide access to the drives. A low-capacity drive tray does not have drawers. For a high-capacity drive tray, you must specify the identifier (ID) of the drive tray, the ID of the drawer, and the ID of the slot in which a drive resides. For a low-capacity drive tray, you need only specify the ID of the drive tray and the ID of the slot in which a drive resides. For a low-capacity drive tray, an alternative method for identifying a location for a drive is to specify the ID of the drive tray, set the ID of the drawer to **0**, and specify the ID of the slot in which a drive resides.

Performance Tuning

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

Monitoring the Performance

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

Table 27. Information About Storage Array Performance

Type of Information	Description
Devices	These devices are included in the file: ■ **Controllers** – The controller in slot A or slot B and a list of the volumes that are owned by the controller ■ **Volumes** – 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 drive
Current KB per second	The current transfer rate in kilobytes per second (current means that the number of kilobytes per second since the last time that the polling interval elapsed, causing an update to occur)
Maximum KB per second	The highest data transfer value that is achieved in the current kilobyte-per-second statistic block
Current I/O per second (IOPS)	The current number of I/Os per second (current means the number of I/Os per second since the last time that the polling interval elapsed, causing an update to occur)
Maximum I/O per second	The highest number of I/Os achieved in the current I/O-per-second statistic block

The command takes this form:

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

In this command, *`filename`* is the name of the file in which you want to save the performance statistics. You can use any file name that 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 these commands to specify how often statistics are collected.

- **set session performanceMonitorInterval**
- **set session performanceMonitorIterations**

The **performanceMonitorInterval** parameter defines the frequency of gathering 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. Use this parameter when you want to change the frequency for gathering performance data.

The **performanceMonitorIterations** 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. Use this parameter when you want to redefine the number of samples to capture.

The polling interval and the number of iterations that you specify remain in effect until you end the session. After you end the session, the polling interval and the number of iterations return to the default values.

Changing the RAID Levels

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

NOTE RAID levels do not apply to dynamic disk pools (DDP). The use of dynamic RAID level migration is valid only when your volume group is currently RAID 6.

To change the RAID level, use this command:

```
set volumeGroup [volumeGroupNumber]
raidLevel=(0 | 1 | 3 | 5 | 6)
```

In this command, *volumeGroupNumber* is the number of the volume group for which you want to change the RAID level.

Alternatively, you can use the command with a volume group name instead of a volume group number, as shown:

```
set volumeGroup [volumeGroupName]
raidLevel=(0|1|3|5|6)
```

Changing the Segment Size

When you create a new volume, you can define the segment size for that volume. In addition, you can change the segment size later to optimize performance. In a multiuser database or file system storage environment, set your segment size to minimize the number of drives that are needed to satisfy an I/O request. Use larger values for the segment size. Using a single drive for a single request leaves other drives 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; use smaller values for the segment size. To change the segment size, use this command:

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

In this command, *segmentSizeValue* is the new segment size that you want to set. Valid segment size values are **8**, **16**, **32**, **64**, **128**, **256**, and **512**.

You can identify the volume by name or by WWID. The volume names must be enclosed in square brackets. If a volume name also has special characters or numbers, you must enclose the volume name in double quotation marks. When using the WWID, enclose the WWID in double quotation marks (" ") inside angle brackets (< >). Do not include colons in the WWID. (For usage information, refer to the **set volume** command in the *Command Line Interface and Script Commands Programming Guide*.)

Changing the Cache Parameters

There are two kinds of cache used with the storage systems:

- Controller cache – A physical memory dedicated to I/O operations between controller and hosts and between controller and disk drives.

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

- **set storageArray**
- **set volume**

The **set storageArray** command lets you change these controller cache settings:

- The controller cache block size
- The controller cache flush start percentage
- The controller cache flush stop percentage

The **set volume** command lets you change these controller cache settings:

- The controller cache flush modifier
- The controller cache without batteries enabled or disabled
- The controller mirror cache enabled or disabled
- The controller read cache enabled or disabled
- The controller write cache enabled or disabled
- The controller cache read ahead multiplier
- The controller cache read Prefetch enabled or disabled

Defragmenting a Volume Group

When you defragment a volume group, you consolidate 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 volume 1 and volume 3, your volume group is configured as follows:

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

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

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

To defragment a volume group, use this command:

```
start volumeGroup [volumeGroupNumber] defragment
```

In this command, *volumeGroupNumber* is the identifier for the volume group.

Troubleshooting and Diagnostics

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

Detailed Error Reporting

Data collected from an error encountered by the CLI is written to a file. Detailed error reporting under the CLI works as follows:

- If the CLI must abnormally end running CLI commands and script commands, error data is collected and saved before the CLI finishes.
- The CLI saves the error data by writing the data to a standard file name.
- The CLI automatically saves the data to a file. Special command line options are not required to save the error data.
- You are not required to perform any action to save the error data to a file.
- The CLI does not have any provision to avoid over-writing an existing version of the file that contains error data.

For error processing, errors appear as two types:

- Terminal errors or syntax errors that you might enter
- Exceptions that occur as a result of an operational error

When the CLI encounters either type of error, the CLI writes information that describes the error directly to the command line and sets a return code. Depending on the return code, the CLI also might write additional information about which terminal caused the error. The CLI also writes information about what it was expecting in the command syntax to help you identify any syntax errors that you might have entered.

When an exception occurs while a command is running, the CLI captures the error. At the end of processing the command (after the command processing information has been written to the command line), the CLI automatically saves the error information to a file.

The name of the file to which error information is saved is `excprpt.txt`. The CLI tries to place the `excprpt.txt` file in the directory that is specified by the system property `devmgr.datadir`. If for any reason the CLI cannot place the file in the directory specified by `devmgr.datadir`, the CLI saves the `excprpt.txt` file in the same directory from which the CLI is running. You cannot change the file name or the location. The `excprpt.txt` file is overwritten

every time that an exception occurs. If you want to save the information in the `excprpt.txt` file, you must copy the information to a new file or a new directory.

Collecting All Support Data

- AutoSupport (ASUP)
- **`save storageArray supportData`** command

The following table lists the type of support data that you can collect. For the commands that you can use to collect support bundle data, refer to *Command Line Interface and Script Commands Programming Guide*.

Table 28. Support Data for the Storage Array

Type of Data	Description and File Name
AutoSupport transmission logs	A detailed list of actions that occur during collection and dispatch of AutoSupport messages. `asup-transmission-logs.txt`
Controller health image metadata	A detailed list of information for the last controller health image captured by a controller. `core-dump-info.xml`
SANtricity Enterprise Management Window configuration	A detailed list of the configuration managed through the EMW portion of SANtricity. `emwdata_v04.bin`
Failed repository analysis	Detailed information about a failed repository. `failed-repository-analysis.txt`
Infiniband interface statistics	A detailed list of performance statistics specific to the Infiniband host interface. `ib-statistics.csv`
Controller-drive error event log	A detailed list of drive error events detected by the controllers. `controller-drive-error-event-log.txt`
SANtricity user preferences	A detailed list of user preferences managed by SANtricity. `perf_01.bin`
SANtricity runtime information	A detailed list of SANtricity runtime information. `msw-runtime-info.txt`
SAS physical layer (SAS PHY)	A detailed list of errors that have been detected in the traffic flow between the devices on the SAS interconnect. `sas-phy-error-logs.csv`
Switch-on-a-chip (SOC) error statistics	Information from the loop-switch ports that are connected to Fibre Channel devices. `soc-statistics.csv`
Cable and connections	A detailed list of actions that describes the drive side cabling and connections. `connection.txt`

Type of Data	Description and File Name
Drive command aging timeout	A detailed list of drive information related to current and default aging timeout values. `drive-command-aging-timeout.txt`
Feature bundle	A detailed description of the settings for all features managed through the feature bundle. `feature-bundle.txt`
Firmware inventory	A detailed list of all of the firmware running on the controllers, the drives, the drawers, and the environmental services modules (ESMs) in the storage array. `firmware-inventory.txt`
I/O path statistics	A detailed list of statistics related to the I/O path. `io-path-statistics.7z`
iSCSI session and connection information	A detailed list of information related to the iSCSI session and connection information for an iSCSI host interface. `iscsi-session-connections.txt`
iSCSI interface statistics	A detailed list of performance statistics specific to the iSCSI host interface. `iscsi-statistics.csv`
Major Event Log	A detailed list of errors that occur on the storage array. The list is stored in reserved areas on the drives in the storage array. The list records configuration events and failures with storage array components. `major-event-log.txt`
NVSRAM	A controller file that specifies the default settings for the controllers. `nvsram-data.txt`
Object bundle	A detailed description of the status of the storage array and its components, which was valid at the time that the file was generated. The object bundle file is a binary file and does not contain human-readable information. `object-bundle.bin`
Performance statistics	A detailed description of how a storage array is performing. Collected data includes the I/O activity of specific controllers or volumes, the transfer rate of the controller, the current I/Os per second, and the maximum I/Os per second. `performanceStatistics.csv` **NOTE** The information about storage array performance is not included in the AutoSupport bundle. You must run the **`save storageArray performanceStats`** command to collect this information.
Persistent reservations and persistent registrations	A detailed list of volumes on the storage array and persistent reservations and persistent registrations. `persistent-reservations.txt`

Type of Data	Description and File Name
Read link status	A detailed list of errors that have been detected in the traffic flow between the devices on the Fibre Channel loop. A file of historical read link status data might also be included in the archive. `read-link-status.csv`
Recovery Guru procedures	A detailed list of recovery guru procedures for current persisting failures. `recovery-guru-procedures.html`
Recovery profile	A detailed description of the latest recovery profile record and historical data. `recovery-profile.csv`
State capture data	A detailed description of the current state of the controllers. `state-capture-data.txt`
Storage array configuration	A detailed listing of the hardware components and the software components that comprise the storage array configuration. `storage-array-configuration.cfg`
Storage array profile	A list of all components and properties of a storage array. `storage-array-profile.txt`
Controller trace buffer	DQ Trace buffer of each controller. `trace-buffers.7z`
Unreadable sectors	A detailed list of all of the unreadable sectors that have been logged to the storage array. `badBlocksData.txt`
Cumulative statistics bundles	A detailed list of bundles of drive-related and volume-related statistical data. `cumulative-statistics-bundles.7z`
Manifest file	A table listing the files that are included in the archive file and collection status of each of those files. `manifest.xml`
X-header data file	AutoSupport message header that consists of plain text key-value pairs that include information about the storage array and message type. `x-header-data.txt`

Collecting Drive Data

To gather information about all of the drives in a storage array, use the **`save allDrives`** command. This command collects sense data and saves the data to a file. The sense data consists of statistical information that is maintained by each of the drives in the storage array. When you have collected the file, send it to your Technical Support Representative.

Diagnosing a Controller

The **`diagnose controller`** command provides these tests that help you make sure that a controller is functioning correctly:

- Loopback drive channel
- Read
- Write
- Data-loopback
- Pattern

The loopback drive channel identifies the drive channels on which you want to run the diagnostic tests. You can either choose to run the diagnostics on all channels or select a specific channel on which to run diagnostics. If you select a specific channel, valid values for the drive channels are **1**, **2**, **3**, **4**, **5**, **6**, **7**, or **8**.

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, and the read test checks 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 it 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, 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 of the other tests pass.

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

The test results contain a generic, overall status message and a set of specific test results. Each test result contains these items:

- Test (read, write, or data loopback)
- Port (read or write)
- Level (internal or external)
- Status (pass or fail)

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

Running Read Link Status Diagnostics

NOTE This service operation and related commands are only for legacy Fibre Channel storage arrays.

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 that are accumulated over time. The counts provide coarse measure of the integrity of the components and devices on the loop. By analyzing the error counts that are retrieved, you can determine the components or devices within the Fibre Channel loop that

might be experiencing problems communicating with the other devices on the loop. A high error count for a particular component or device indicates that it might 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 sequence 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 RLS diagnostics:

- **`reset storageArray RLSBaseline`** – Resets the RLS baseline for all devices by setting all of the counts to 0.
- **`save storageArray RLSCounts`** – Saves the RLS counters to a file that you can review later. The default file name is `readLinkStatus.csv`.

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

The following table lists the type of data contained by the file that is generated by the **`save storageArray RLSBaseline`** command.

Table 29. RLS Baseline Data for the Storage Array

Type of Data	Description
Devices	A list of all devices on the Fibre Channel loop. The devices appear in channel order. Within each channel, the devices are sorted according to the device position in the loop.
Baseline time	The date and time when the baseline was set.
Elapsed time	The time that has elapsed from when the baseline time was set to when the read link status was gathered.
Invalid transmission word (ITW)	The total number of ITW errors that were detected on the Fibre Channel loop from the baseline time to the current date and time. ITW might also be referred to as the Received Bad Character Count. 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. This data is the key error count to be used when analyzing the error count data.
Link failure (LF)	The total number of LF errors that were detected on the Fibre Channel loop from the baseline time to the current date and time. 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.
Loss of synchronization (LOS)	The total number of LOS errors that were detected on the Fibre Channel loop from the baseline time to the current date and time. 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 increases.
Loss of signal (LOSG)	The total number of LOSG errors that were detected on the Fibre Channel loop from the baseline date to the current date and time. LOSG errors typically indicate a loss of signal from the transmitting node or the physical component within the Fibre Channel loop. Physical components where a loss of signal typically occurs include the gigabit interface converters (GBICs), the Small Form-factor Pluggable (SFP) transceivers, and the Fibre Channel fiber-optic cable.

Type of Data	Description
Primitive sequence protocol (PSP)	The total number of PSP errors that were detected on the Fibre Channel loop from the baseline date to the current date and time. PSP refers to the number of N_Port protocol errors that were detected and Link Reset Response (LRR) primitive sequences that were received while the link is up. An LRR is issued by another N_Port in response to a link reset. An N_Port is a Fibre Channel-defined port at the end of a link, such as a server or a 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 N_Port identifier. If an N_Port is connected to a loop, it becomes an NL_Port. An NL_Port is a Fibre Channel controller ID in a hexadecimal number. The hexadecimal number varies depending on the topology: ■ For a private arbitrated loop, the ID is a 1-byte arbitrated loop physical address (ALPA). ■ For all other arbitrated loops, it appears as a single 24-bit hexadecimal number (a triplet of domain, area, and ALPA where each field is 1 byte). ■ For fabric and point-to-point, the ID is a 3-byte hexadecimal number used in the DID and SID (destination identifier and source identifier) fields of Fibre Channel frames.
Invalid cyclic redundancy check (ICRC)	The total number of ICRC errors that were detected on the Fibre Channel loop from the baseline date to the current date and time. 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 characters and the old characters do not match, an error is posted, and the data is re-sent.

Interpreting the RLS Results

The way that you interpret the RLS results is based on the concept that the device immediately following the problematic component will have the largest number of invalid transition word (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 discrepancy into consideration.

Recovery Operations

Recovery operations include repairing the storage array and returning it to an operational state. This might involve replacing a failed canister, a failed controller, a failed drive, restoring data, or changing the operational mode of the storage array. For information about when it is appropriate to replace a canister, see Replacing Canisters.

Setting the Controller Operational Mode

A controller has three operational modes:

- Online
- Offline
- Service

Placing a controller online sets it to the Optimal state 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 is 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.

ATTENTION Possible loss of data access – Placing a controller offline can cause loss of data.

Use Service mode to replace canisters, such as 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 preferred path of the volume group. This action might significantly reduce performance. The volume groups are automatically 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 causes I/O errors unless a multi-path driver is installed on the host. Before you place a controller in Service mode, make sure that the volumes are not in use, or a multi-path driver is installed on all of the hosts that are using these volumes.

In addition, if you do not have a multi-path driver, you must make appropriate operating system-specific modifications to make sure that the volume groups moved are accessed on the new path when you change to Service mode.

IMPORTANT Place a controller in Service mode only under the direction of a Technical Support Representative.

To change the operational mode of a controller, use this command:

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

Changing the Controller–Volume Ownership

You can change which controller is the owner of a volume by using the **`set volume`** command. The command takes this form:

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

You can identify the volume by name or by WWID. The volume names must be enclosed in square brackets. If a volume name also has special characters or numbers, you must enclose the volume name in double quotation marks. When using the WWID, enclose the WWID in double quotation marks (" ") inside angle brackets (< >). Do not include colons in the WWID. (For usage information, refer to the **`set volume`** command in the *Command Line Interface and Script Commands Programming Guide*.)

Initializing a Drive

ATTENTION Possible loss of data access – When you initialize a drive, all data on the drive is lost.

You must initialize a drive when you have moved a drive that was previously part of a multidisk volume group from one storage array to another. If you do not move the entire set of drives, the volume group information and the volume information on the drives that you move are incomplete. Each drive that you move contains only part of the information that is defined for the volume and the volume group. To be able to reuse the drives to create a new volume group and volume, you must delete all of the old information from the drives by initializing the drive.

When you initialize a drive, all of the old volume group information and volume information are deleted, and the drive is returned to an unassigned state. Returning a drive 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 drive, use this command:

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

In this command, *`trayID`*, *`drawerID`*, and *`slotID`* are the identifiers for the drive. These identifiers support both high-capacity drive trays and low-capacity drive trays. A high-capacity drive tray has drawers that hold the drives. The drawers slide out of the drive tray to provide access to the drives. A low-capacity drive tray does not have drawers. For a high-capacity drive tray, you must specify the identifier (ID) of the drive tray, the ID of the drawer, and the ID of the slot in which a drive resides. For a low-capacity drive tray, you need only specify the ID of the drive tray and the ID of the slot in which a drive resides. For a low-capacity drive tray, an alternative method for identifying a location for a drive is to specify the ID of the drive tray, set the ID of the drawer to **`0`**, and specify the ID of the slot in which a drive resides.

Reconstructing a Drive

If two or more of the drives in a volume group have failed, the volume shows a status of Failed. All of the volumes in the volume group are no longer operating. To return the volume group to an Optimal status, you must replace the failed drives. Then, you must reconstruct the data on the new drives. The data that you reconstruct is the data as it would appear on the failed drives.

IMPORTANT You can use this command only when the drive is assigned to a disk pool or to a RAID level 1, RAID level 3, RAID level 5, or RAID level 6 volume group.

To reconstruct a drive, use this command:

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

In this command, *`trayID`*, *`drawerID`*, and *`slotID`* are the identifiers for the drive. These identifiers support both high-capacity drive trays and low-capacity drive trays. A high-capacity drive tray has drawers that hold the drives. The drawers slide out of the drive tray to provide access to the drives. A low-capacity drive tray does not have drawers. For a high-capacity drive tray, you must specify the identifier (ID) of the drive tray, the ID of the drawer, and the ID of

the slot in which a drive resides. For a low-capacity drive tray, you need only specify the ID of the drive tray and the ID of the slot in which a drive resides. For a low-capacity drive tray, an alternative method for identifying a location for a drive is to specify the ID of the drive tray, set the ID of the drawer to **0**, and specify the ID of the slot in which a drive resides.

Initializing a Volume

ATTENTION Possible loss of data access – When you initialize a volume, all data on the volume and all of the information about the volume are destroyed.

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

Consider these restrictions when you initialize a volume:

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

To initialize a volume, use this command:

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

In this command, *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 a 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. That controller becomes the current owner of the volumes. This change is considered to be a routine ownership change and is reported in the Event Log.

To redistribute volumes to their preferred controllers, use this command:

```
reset storageArray volumeDistribution
```

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

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

Replacing Canisters

Beginning with the CE6998 controller tray, components, such as the controller canisters and the power-fan canisters, have a Service Action Allowed indicator light. This indicator light is a blue LED. The Service Action Allowed indicator light helps to make sure that you do not remove a canister before it is safe to do so.

ATTENTION Possible loss of data access – Never remove a component that has a Service Action Required indicator light on unless the Service Action Allowed indicator light is on.

If a component fails and must be replaced, the Service Action Required indicator light on that canister comes on to indicate that service action is required, provided no data availability dependencies or other conditions exist that dictate the canister should not be removed. The Service Action Allowed indicator light automatically comes on or goes off when conditions change. In most cases, the Service Action Allowed indicator light comes on steadily when the Service Action Required indicator light comes on for the canister.

The ability to remove a canister depends on the data availability dependencies of the controller tray or the controller-drive tray. The Service Action Allowed indicator light does not come on if removing a canister jeopardizes data on the drive trays or current I/O activity. An example of limiting when you can remove a canister is when one controller canister has a Service Action Required indicator light on. You cannot remove the other controller canister (the Service Action Allowed indicator light does not come on), because doing so would jeopardize the data either on the drive trays or transitioning through the controllers.

A less obvious example is when the power supply for the controller canister in slot A has failed, and the controller canister in slot B has failed. Removing the controller canister in slot B before replacing the failed power-fan canister causes the controller canister in slot A to lose power, which results in a loss of data access. This action occurs because power distribution from each power-fan canister is through the controller canister that is physically connected to that power-fan canister.

So, in the preceding example, these actions occur:

- The power-fan canister has both its Service Action Required indicator light and its Service Action Allowed indicator light on.
- The controller canister in slot B has only its Service Action Required indicator light on, but its Service Action Allowed indicator light is off.
- After the failed power-fan canister has been replaced, the Service Action Allowed indicator light comes on for the controller canister in slot B.

The following table shows when the Service Action Allowed indicator light does not come on for each canister (the indicator light is suppressed). An **X** in a table cell indicates that service is not allowed, therefore the Service Action Allowed light does not come on. For example, if the power supply in the power-fan canister in slot A has failed, then replacement of the controller canister in slot B, the interconnect-battery canister, or the power-fan canister in slot B is not allowed, which is indicated when the Service Action Allowed indicator light stays off for those canisters.

Table 30. Service Action Not Allowed

Description of Failure or Circumstance	Canister Description			
	Controller in Slot A	Controller in Slot B	Power-Fan in Slot A	Power-Fan in Slot B
The controller canister in slot A has failed or is locked down.		X		
The controller canister in slot B has failed or is locked down.	X			

Description of Failure or Circumstance	Canister Description			
	Controller in Slot A	Controller in Slot B	Power-Fan in Slot A	Power-Fan in Slot B
The controller canister in the slot A drive path is unavailable.		X		
The controller canister in the slot B drive path is unavailable.	X			
The controller canister in slot A has been removed.		X		
The controller canister in slot B has been removed.	X			
The power supply in the power-fan canister in slot A has failed.				X
A fan in the power-fan canister in slot A has failed or has no input power.				X
The power-fan canister in slot A has been removed.				X
The power supply in the power-fan canister in slot B has failed.			X	
A fan in the power-fan canister in slot B has failed or has no input power.			X	
The power-fan canister in slot B has been removed.			X	

Appendix A - Example Script Files

This appendix provides example scripts for configuring a storage array. These examples show how the script commands appear in a complete script file. Also, you can copy these scripts and modify them to create a configuration unique to your 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 other storage arrays. You can also use this file to restore an existing configuration that has become corrupted. You also can copy an existing file to serve as a 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 by using a text editor, such as Microsoft Notepad. The maximum line length is 256 characters. The command syntax must conform to the guidelines in Chapter 1 - About the Command Line Interface and Chapter 2 - About the Script Commands. When you create a new script file, you can use any file name and extension that will run on the host operating system.

This example shows how to run a script file from the command line.

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-f scriptfile.scr;
```

Configuration Script Example 1

This example creates a new volume by 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 cacheReadPrefetch=TRUE 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 might want to include blank lines in your script files to separate blocks of commands or make a comment that stands out. To include a comment, enter 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 shows text that is bounded by double quotation marks (“ ”) on a display monitor when the script file runs. In this example, the text `Create RAID 5 Volume 7 on existing Volume Group 1` serves as a title that describes the expected results of running this script file.

The line beginning with `//Create` is a comment that explains that the purpose of this script file is to create a new volume by using the **`create volume`** command on an existing volume group.

The line beginning `//Note:` is a comment in the script file that explains that the size of the last volume created that uses all of the available capacity because the `capacity` parameter is not used.

The command in this example creates a new volume in volume group 1. The volume has RAID level 5. The volume name (user label) is 7. (Note the double quotation marks around the 7. The double quotation marks define that the information in the double quotation marks is a label.) The new volume is assigned to the controller in slot A in the controller tray. The segment size is set to 16. The volume has a read ahead multiplier value of 256. The capacity of the volume is 2 GB.

The command takes this form:

```
create volume volumeGroup=volumeGroupNumber
userLabel=volumeName
[freeCapacityArea=freeCapacityIndexNumber]
[capacity=volumeCapacity | owner=(a | b) |
cacheReadPrefetch=(TRUE | FALSE) |
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 showing “`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 are now set.

The **`set volume`** 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, as in this example:

```
set volume[“7”] cacheFlushModifier=10
cacheWithoutBatteryEnabled=false
modificationPriority=high;
```

By using separate lines, you can see more clearly the parameters that you are setting and the values to which you are setting the parameters. Blocking the parameters in this manner makes it easier to either edit the file or copy specific parameter settings for use in another script file.

Configuration Script Example 2

This example creates a new volume by using the **`create volume`** command with user-defined drives in the storage array.

```
Show “Create RAID3 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 volume 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 voluem 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”] redundantCheckEnabled=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 you can define specific drives to include in the volume. Use the **`show storageArray profile`** command to find out what drives are available in a storage array.

The **`create volume`** command takes this form:

```
create volume raidLevel=(0 | 1 | 3 | 5 | 6) userLabel=volumeName
drives=(trayID1,slotID1...trayIDn,slotIDn)
[capacity=volumeCapacity | owner=(a | b) |
cacheReadPrefetch=(TRUE | FALSE) |
segmentSize=segmentSizeValue]
[trayLossProtect=(TRUE | FALSE)]
```

Appendix B - Asynchronous Write Mode Mirror Utility

This appendix describes the host utility to achieve periodic consistency with Asynchronous Write Mode Mirror configurations. This appendix also describes how to run the Asynchronous Write Mode utility.

NOTE The Asynchronous Write Mode Mirror utility works only with the synchronous remote mirror commands. This utility does not work with the asynchronous remote mirror commands.

Description of the Asynchronous Write Mode Mirror Utility

The Asynchronous Write Mode Mirror utility enables you to periodically synchronize the Synchronous Mirroring pairs in your storage array. When defining a Synchronous Mirroring configuration, you have the option to set the write modes to either Synchronous or Asynchronous. 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. Asynchronous write mode offers faster host I/O performance, but it does not guarantee that a copy operation has successfully completed before processing the next write request. With Asynchronous write mode, you cannot make sure that a volume, or collection of volumes, at a secondary site ever reach a consistent, recoverable state.

The Asynchronous Write Mode Mirror utility enables you to bring a collection of asynchronous remote volumes into a mutually consistent and recoverable state. You can choose to run the utility based on application demands, link state and speed, and other factors that are relevant to your environment.

The Asynchronous Write Mode Mirror utility has these characteristics:

- The utility is implemented as a command line-invoked Java-based application.
- The utility is bundled as part of the SANtricity Storage Manager installation package.
- The utility accepts a command line argument that lets you specify the name of a configuration file that contains a complete specification of the work to be carried out by the utility.
- More than one instance of the utility can run concurrently, as long as the utilities do not try to process any of the same volumes and mirrors.

NOTE The Asynchronous Write Mode Mirror utility does not check to make sure that concurrently running instances of the utility are not trying to process the same volumes and mirrors. If you choose to simultaneously run more than one instance of the Asynchronous Write Mode Mirror utility, you must make sure that the configuration files that you choose to run do not list the same volumes and mirrors.

Operation of the Asynchronous Write Mode Mirror Utility

The Asynchronous Write Mode Mirror utility performs steps that generate a recoverable state for multiple mirror volumes at a secondary site. The utility runs these steps to create consistent, recoverable images of a set of volumes:

1. **On the primary storage array** – The utility reconfigures all of the participating volumes from asynchronous write mode mirroring to synchronous write mode mirroring. This action makes sure that the stream of write operations becomes recoverable on the secondary side.

2. **On the primary storage array** – The utility polls all of the participating volumes until the associated mirror states all have the Optimal state. In cases where the remote link is slow or the primary host I/O activity is high, one or more mirrors are likely to be in the Unsynchronized state before they transition to the Synchronized state. By waiting until all of the mirrors have Optimal status, the utility makes sure that all of the delta logs for the affected volumes are cleared, and the secondary volumes are recoverable.
3. **On the primary storage array** – The utility suspends the mirrored pairs for all of the participating volumes. This action causes updates to stop on the secondary side, leaving the secondary volumes in a recoverable state because they were being updated in Synchronous mode immediately before the suspension. By separating the mirrors in this manner, the primary-side applications run faster, while leaving the secondary volumes in a recoverable state. The delta log tracks changes made because of application writes on the primary side while in this state.
4. **On the secondary storage array** – The utility generates a snapshot (legacy) of each participating volume on the secondary side, which creates point-in-time images that are recoverable.
5. **On the primary storage array** – The utility resumes the mirroring operations for all of the participating volumes. This action causes the mirrors to transition to the Synchronized state and start the process of restoring coherency between the primary site and the secondary site.
6. **On the primary storage array** – The utility reconfigures all of the affected volumes for Asynchronous Write Mode.

Running the Asynchronous Write Mode Mirror Utility

The Asynchronous Write Mode Mirror utility uses a command line argument that enables you to specify the name of a configuration file. The configuration file contains a complete specification of the input parameters that are needed by the utility. To run the utility, enter this syntax:

```
asyncRVMUtil configuration_file -d debug_file
```

In this command, *`configuration_file`* is the file that you provide as input. The configuration file specifies the Synchronous Mirroring volumes that you want to synchronize by using the utility. When you create the configuration file, use these conditions to define the volumes in the file:

- All the primary volumes in a volume set must belong to the same storage array.
- The maximum number of volume sets that you can specify in the file is four.
- The maximum number of mirrored pairs that you can specify as part of a consistency group is eight.

The optional parameter, `-d`, lets you specify a file to which you can send information regarding how the utility runs. In this example, the file name is *`debug_file`*. The debug file contains trace information that can be reviewed by your Technical Support representative to determine how well the Asynchronous Write Mode Mirror utility has run.

NOTE Depending on the location of the configuration file and the debug file, you must specify the complete path with the file name.

To run the Asynchronous Write Mode Mirror utility, you must enter the **`asyncRVMUtil`** command from the command line. Because UNIX operating systems are case sensitive, you must type the command exactly as shown. On Windows operating systems, you can type the command in all uppercase, in all lowercase, or in mixed case.

NOTE To use the Asynchronous Write Mode Mirror utility, you must be managing the storage array by using the command line interface, not the graphical user interface of SANtricity Storage Manager.

Configuration Utility

The configuration file is an ASCII flat text file that provides the information for the Synchronous Mirroring synchronization used by the Asynchronous Write Mode Mirror utility. The file defines the mirror volume sets to be synchronized. All of the mirror volumes in the volume sets defined in the configuration file are run collectively to create a recoverable image. If any one of the mirrors in the volume set fails, the operation is stopped for this volume set and carried on to the next volume set that is listed in the configuration file.

The configuration file supports this syntax:

```
content ::= {spec}
spec ::= logSpec | volumeSetSpec

logSpec ::= "Log" "{" {logAttribute} "}"
logAttribute ::= fileSpec
fileSpec ::= "file" "=" fileName

volumeSetSpec ::= "VolumeSet" volumeSetName
"{" {volumeSetAttribute} "}"
volumeSetAttribute ::= timeoutSpec | mirrorSpec

timeoutSpec ::= "OptimalWaitTimeLimit" "=" integer

mirrorSpec ::= "Mirror" "{" {mirrorAttribute} "}"
mirrorAttribute ::= primarySpec | secondarySpec |
snapshotSpec

primarySpec ::= "Primary" "=" volumeSpec
secondarySpec ::= "Secondary" "=" volumeSpec
snapshotSpec ::= "Copy" "=" volumeSpec
volumeSpec ::= storageArrayName"."volumeUserLabel
```

In this syntax, items enclosed in double quotation marks (“ ”) are terminal symbols. Items separated by a vertical bar (|) are alternative values (enter one or the other, but not both). Items enclosed in curly braces ({ }) are optional (you can use the item zero or more times).

These definitions are provided for non-terminals in the syntax:

- *integer* – The timeout value must be an integer (decimal digits from 0–9).
- *volumeSetName* – The name of the set of volumes on which you want to run the Asynchronous Write Mode Mirror utility.
- *fileName* – The name of a file, using characters and conventions that are appropriate for the system on which the application is running.
- *storageArrayName* – The label that you have assigned for a storage array, as would be used in the CLI to specify the name of the storage array.
- *volumeUserLabel* – The label that you have assigned for a volume that uniquely identifies the volume within the storage array.

NOTE Names and labels can be any characters that are defined as appropriate for your operating system. The maximum length for a name or label is 30 characters. If the name or label contains special characters (as defined by the operating system) or period characters, you must enclose the name or label in double quotation marks (“ ”). You can, optionally, enclose the name or label in double quotation marks at any time.

These items are considered syntax errors:

- More than one `logSpec` command in the input file
- Zero or more than one `fileSpec` attribute in a `logSpec` command (you must include exactly one `fileSpec` attribute in the `logSpec` command)
- More than one `timeoutSpec` attribute in a `volumeSetSpec` command
- Zero or more than one `primarySpec` attribute in a `mirrorSpec` command (you must include exactly one `primarySpec` attribute in the `mirrorSpec` command)
- Zero or more than one `secondarySpec` attribute in a `mirrorSpec` command (you must include exactly one `secondarySpec` attribute in the `mirrorSpec` command)
- Zero or more than one `snapshotSpec` attribute in a `mirrorSpec` command (you must include exactly one `snapshotSpec` attribute in the `mirrorSpec` command)

IMPORTANT In the Asynchronous Write Mode Mirror utility configuration file, you must specify the primary volume, the secondary volume, and the copy volume. The utility does not make sure that the secondary volume is correct for the Synchronous Mirroring relationship. The utility also does not make sure that the snapshot (legacy) volume is actually a snapshot (legacy) for the secondary volume. *You must make sure that these volumes are correct.* If the volumes are not correct, the utility will run, but the volumes will not be consistent. For each mirror, the secondary volume and the copy volume must reside on the same storage array.

This example shows a configuration file for the Asynchronous Synchronous Mirroring utility.

```
Log{ file="d:\rvm-consistency.log" }
VolumeSet "set1" {
 optimalWaitTimeLimit = 15
 Mirror {
  Primary = LosAngelesArray.PayrollVolume
  Secondary = NewYorkArray.PayrollVolume
  Copy = NewYorkArray.PayrollVolumeImage
}
 Mirror {
  Primary = LosAngelesArray.PayrollVolume
  Secondary = BostonArray.PayrollVolume
  Copy = BostonArray.PayrollVolumeImage
 }
}

VolumeSet "set2" {
 Mirror {
  Primary = BostonArray.HRVolume
  Secondary = LosAngelesArray.HRVolume
  Copy = LosAngelesArray.HRVolumeImage
 }
}
```

Appendix C - Simplex-to-Duplex Conversion

Some models of controller trays and controller-drive trays are available in either a simplex configuration (one controller) or a duplex configuration (two controllers). You can convert a simplex configuration to a duplex configuration by installing new nonvolatile static random access memory (NVSRAM) and a second controller. This appendix explains how to convert a simplex configuration to a duplex configuration by using CLI commands or by using the storage management software.

Simplex-to-Duplex Conversion

General Steps

To access this product, go to the Support Site at Quantum.com

You can upgrade a controller tray or a base system that has a simplex configuration to a duplex configuration by performing these tasks:

1. Install new NVSRAM on the existing controller in your controller tray or base system.
2. Revise the controller tray configuration or the base system configuration to run with two controllers.
3. Install a second controller.
4. Connect the host cables.
5. Connect the expansion unit cables.
6. Run diagnostics to make sure that your new configuration is running correctly.

Because the instructions in this procedure apply to different models of controllers, this procedure does not describe in detail how to install any one controller. For information about latches and ports, refer to the removal and replacement instruction for your controller. For information about cabling, refer to the *Hardware Cabling Guide*.

Tools and Equipment

The procedures in this appendix require these items:

- Antistatic protection
- A second controller
- Small Form-factor Pluggable (SFP) transceivers (for Fibre Channel configurations)
- Host-to-controller cables
- Controller-to-environmental services module (ESM) cables
- The removal and replacement instruction for your model of controller
- The *Hardware Cabling Guide*

Step 1 – Installing the Duplex NVSRAM

IMPORTANT Before trying to download NVSRAM, you must contact your Technical Support Representative to make sure that you are downloading the NVSRAM that is appropriate for the controller in your storage array.

NVSRAM files specify the default settings for the controller tray controllers or base system controllers. Follow the instructions in this step to upgrade the NVSRAM on the controller in your controller tray or your base system.

To get a copy of the latest NVSRAM, perform one of these tasks:

- Download the duplex NVSRAM by using the command line interface.
- Download the duplex NVSRAM by using the graphical user interface (GUI) of the storage management software.

Make sure that the controller tray or the base system has an Optimal status. If one or more managed devices has a Needs Attention status, determine and correct the condition that created the Needs Attention status before proceeding with this conversion instruction.

Downloading the NVSRAM by Using the Command Line Interface

1. Make a copy of your storage array profile, and save it in the event that you might need to restore the storage array.
2. Start the command line interface.
3. On the command line, type this command, and press **Enter**. In this command, *ctlr-A_IP_address* is the IP address of the original simplex controller, and *filename* is the complete file path and name of the file that contains the new NVSRAM. Valid file names must end with a `.dlp` extension. Enclose the file name in double quotation marks (“ ”).

   ```
   smcli ctlr-A_IP_address -c “download storageArray NVSRAM file=“filename”;”
   ```

Downloading the NVSRAM by Using the GUI

1. Make a copy of your storage array profile, and save it in the event that you might need to restore the storage array.
2. At the storage management station, start the SMclient software.
3. In the Array Management Window, select **Upgrade** > **Controller NVSRAM**.
4. In the Download NVSRAM dialog, enter the NVSRAM file name in the **Selected NVSRAM** text box. If you do not know the file name, click **Browse**, and navigate to a folder with the NVSRAM files.
5. Select the file that corresponds to your storage array type.
6. Click **OK**.

   The **Confirm Download** dialog appears.
7. To start the download, click **Yes**.
8. Based on the dialog that appears after the download has completed, perform one of these actions:
   - **Download Successful dialog** – Click **Done**.
   - **Error dialog** – Read the information in the dialog, and take the appropriate action.

Step 2 – Setting the Configuration to Duplex

After rebooting the controller tray or the base system, an “alternate controller missing” error message appears. This message indicates that the controller in slot A has successfully converted to duplex mode. This message persists until you have completed the tasks to install the second controller, installed the host cables, and installed the expansion unit cables.

1. Start the command line interface.
2. On the command line, type this command, and press **Enter**. In this command, *ctlr-A_IP_address* is the IP address of the of the original simplex controller.

   ```
   smcli ctlr-A_IP_address -c “set storageArray redundancyMode=duplex;”
   ```

3. Turn the power off to the base system, then turn the power on.

Step 3 – Installing the Second Controller

ATTENTION Possible hardware damage – To prevent electrostatic discharge damage to the tray, use proper antistatic protection when handling tray components.

IMPORTANT For best operation, the new controller must have a part number identical to the existing controller, or the new controller must be a certified substitute. The part number is on a label on the controller. To provide full functionality in dual-controller configurations, make sure that both controllers in the controller tray or the controller-drive tray have the same memory capacity. Although you can install two controllers of different memories in a controller tray or a base system, the mismatch disables some functions, such as cache mirroring.

1. Put on antistatic protection.

   ATTENTION Possible damage to the controller – Do not remove the electrostatic protection until you have finished installing the controller and you have connected the host cables and the expansion unit cables.

2. Unpack the new controller.

   ATTENTION Possible damage to the controller – Bumping the controller against another surface might damage the data connectors on the rear of the controller. Use caution when handling the controller.

3. Remove the blank controller canister from the tray by releasing the levers, and pulling the blank controller canister out of the tray.
4. Slide the new controller canister all of the way into the empty slot in the tray. Rotate the release levers on the controller canister into the closed position to lock the controller canister in place.

Step 4 – Connecting the Host Cables

The steps in this procedure describe how to attach Fibre Channel host cables. The steps for connecting other types of host cables are similar, but they do not require the installation of Small Form-factor Pluggable (SFP) transceivers. For information about host cabling, refer to the *Hardware Cabling Guide*.

1. If there is a black plastic plug in the host port, remove it.
2. Install an SFP transceiver into the controller by pushing the SFP transceiver into the host port until it snaps into place.

ATTENTION Possible degraded performance – To prevent degraded performance, do not twist, fold, pinch, or step on fiber-optic cables. Do not bend fiber-optic cables tighter than a 5-cm (2-in.) radius.

3. Plug one end of the fiber-optic cable into the SFP transceiver in the host port.
4. Plug the other end of the fiber-optic cable into one of the HBAs in the host (direct topology) or into a switch (switch topology).
5. Attach a label to each end of the cable by using this scheme. A label is very important if you need to disconnect the cables later to service a controller.
   - The host name and the host bus adapter (HBA) port (if direct topology)
   - The switch name and port (if switch topology)
   - The controller ID (for example, controller A)
   - The host channel ID (for example, host channel 1)

   Example label abbreviation – Assume that a cable is connected between port 1 in HBA 1 of a host named Engineering and host channel 1 of controller A. A label abbreviation could be as follows:

   ```
   Heng-ABA1/P1, CtA-Hch1
   ```
6. Repeat step 1 through step 5 for each host channel that you intend to use.

Step 5 – Connecting the Controller to a Expansion Unit

The steps in this procedure describe how to attach Fibre Channel cables to a expansion unit. The steps for connecting other types of expansion unit cables are similar, but they do not require the installation of SFP transceivers. For information about connecting the controller to the expansion unit, refer to the *Hardware Cabling Guide*.

1. If there is a black plastic plug in the drive port of the new controller canister, remove it.
2. Insert an SFP transceiver into the drive port on a controller canister.
3. Plug one end of the cable into the SFP transceiver.
4. Plug the other end of the cable into the appropriate in port or out port on the environmental services module (ESM) in the expansion unit as applicable for your cabling configuration.
5. Attach a label to each end of the cable by using this scheme. A label is very important if you need to disconnect the cables later to service a controller.
   - The controller ID (for example, controller A)
   - The drive channel number and port ID (for example, drive channel 1, port 2)
   - The ESM ID (for example, ESM A)
   - The ESM port ID (for example, In, Out, 1A, or 1B)
   - The expansion unit ID

   Example label abbreviation – Assume that a cable is connected between drive channel 1, port 2 of controller A to the out port of the left ESM (A) in expansion unit 1. A label abbreviation could be as follows:

   ```
   CtA-Dch1/P2, Dm1-ESM_A (left), Out
   ```
6. Repeat step 1 through step 5 for each expansion unit.
7. Remove the antistatic protection.

Step 6 – Bringing the Controller Online

1. Use either the CLI (first bullet) or the GUI (second bullet) to bring the controller online.
   - Run the following command:

     ```
     smCLI <DNS-network-name-or-IP-address> -c “set controller [(a | b)]
     availability=online”;
     ```

   - From the Hardware pane in the Array Management Window, right-click the picture of the controller, and select **Advanced > Place > Online**.
2. Using the LEDs on the storage array and information provided by the storage management software, check the status of all trays in the storage array.
3. Does any component have a Needs Attention status?
   - **Yes** – Click the **Recovery Guru** toolbar button in the Array Management Window, and complete the recovery procedure. If a problem is still indicated, contact your Technical Support Representative.
   - **No** – Go to step 4.
4. Use either the CLI (first bullet) or the GUI (second bullet) to distribute the volumes between the controllers.
   - Run the following command:

     ```
     smCLI <DNS-network-name-or-IP-address> -c “set volumes ["volumeName1"
     ... "volumeNameN"] owner=[(a | b)]”;
     ```

   - In the Array Management Window, select **Storage > Volume > Advanced > Redistribute Volumes**.
5. Create, save, and print a new storage array profile.