Command Line Interface and Script Commands Quick Reference

Version 10.77

May 2011

51336-00, Rev. A

Revision History

Version and Date	Description of Changes
51336-00, Rev. A, May 2011	Initial release of the document.

It is the policy of LSI and Sun to improve products as new technology, components, software, and firmware become available. We reserve the right to make changes to any products herein at any time without notice. All features, functions, and operations described herein may not be marketed in all parts of the world. In some instances, photographs and figures are of equipment prototypes. Therefore, before using this document, consult your sales representative or account team for information that is applicable and current. WE DO NOT ASSUME ANY RESPONSIBILITY OR LIABILITY FOR THE USE OF ANY PRODUCTS DESCRIBED HEREIN EXCEPT AS EXPRESSLY AGREED TO IN WRITING BY LSI .

LSI products are not intended for use in life-support appliances, devices, or systems. Use of any LSI product in such applications without written consent of the appropriate LSI officer is prohibited.

LSI Corporate Headquarters	**Email**	**Website**
Milpitas, CA 800-372-2447	globalsupport@lsi.com	www.lsi.com

Document Number: 51336-00, Rev. A

Table of Contents

About this Document

This document provides a complete listing of all of the commands and the syntax for those commands that you need to configure and maintain a storage array. Information about how to configure and maintain a storage array using a command line interface is in *Configuring and Maintaining a Storage Array using the Command Line Interface*

This document supports host software version 10.77 and firmware version 7.77.

Chapter 1: Formatting the Commands

The command line interface (CLI) is a software application that provides a way 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.

The script commands configure and manage a storage array. The script commands are distinct from the CLI commands. 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. When you run a file of script commands, you embed the file name in the CLI command.

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)
`{ ... }` (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

```
SMcli host-name-or-IP-address [host-name-or-IP-address]
[-c "command; {command2};"]
[-n storage-system-name | -w wwID]
[-o outputfile] [-p password] [-e ] [-S ] [-quick]
SMcli host-name-or-IP-address [hostname-or-IP-address]
[-f scriptfile]
[-n storage-system-name | -w wwID]
[-o outputfile] [-p password] [-e] [-S] [-quick]
SMcli (-n  storage-system-name | -w wwID)
[-c "command; {command2};"]
[-o outputfile] [-p password] [-e] [-S] [-quick]
SMcli (-n storage-system-name   -w wwID)
[-f scriptfile]
[-o outputfile] [-p password] [-e] [-S] [-quick]
SMcli -a email: email-address [host-name-or-IP-address1
[host-name-or-IP-address2]]
[-n storage-system-name | -w wwID | -h host-name]
[-I information-to-include] [-q frequency] [-S]
SMcli -x email: email-address [host-name-or-IP-address1
[host-name-or-IP-address2]]
[-n storage-system-name | -w wwID | -h host-name] [-S]
SMcli (-a | -x) trap: community, host-name-or-IP-address
[host-name-or-IP-address1 [host-name-or-IP-address2]]
[-n storage-system-name | -w wwID | -h host-name] [-S]
SMcli -d [-w] [-i] [-s] [-v] [-S]
SMcli -m host-name-or-IP-address -F email-address
[-g contactInfoFile] [-S]
SMcli -A [host-name-or-IP-address [host-name-or-IP-address]]
[-S]
SMcli -X (-n storage-system-name | -w wwID | -h host-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. ■ `supportBundle` – The event and support bundle information 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 contents 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`	Specifies the password for the storage array on which you want to run commands. A password is not necessary under these conditions: ■ A password has not been set on the storage array. ■ The password is specified in a script file that you are running. ■ You specify the password by using the `-c` terminal and this command: `set session password=`*`password`*

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.
`-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.

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 | allobject-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 1 Script Command Object Type Identifiers

Object Type	Identifier
`controller`	a or b
`drive`	Module ID and slot ID
`replacementDrive`	Module 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
`remoteMirror`	Primary volume user label
`snapshot`	Volume user label
`storageArray`	Not applicable
`module`	Module ID

Object Type	Identifier
`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
`pool`	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 *`poolName`*).

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 pool definitions), the actual number of commands is extensive. The commands, however, fall into general categories that are reused when you apply the commands to the different 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 2 General Form of the Script Commands

Syntax	Description
`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.

Syntax	Description

`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.

`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 operation by using an existing snapshot volume. You can change the parameters before the operation is restarted.

`remove` *`object`*
`{`*`statement-data`*`}`
	Removes a relationship from between objects.

`repair` *`object`*
`{`*`statement-data`*`}`
	Repairs errors found by the `check` command.

`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. The Table 3 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)
{ ... } (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 3 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-moduleID	(0-99)
slotID	(1-32)
portID	(0-127)
drive-spec	*moduleID,slotID* or *moduleID,drawerID,slotID* A drive is defined as two or three interger literal values separated by a comma. Low-density modules require two values. High-density modules, those modules that have drawers, require three values.
drive-spec-list	*drive-spec drive-spec*
moduleID-list	*moduleID* {*moduleID*}
esm-spec-list	*esm-spec* {*esm-spec*}
esm-spec	*moduleID,* (left | right)
hex-literal	*0xhexadecimal-literal*
pool-number	*integer-literal*
filename	*string-literal*
error-action	(stop | continue)

Recurring Syntax	Syntax Value
drive-channel-identifier (four drive ports per module)	(1 | 2 | 3 | 4)
drive-channel-identifier (eight drive ports per module)	(1 | 2 | 3 | 4 | 5 | 6 | 7 | 8)
drive-channel-identifier-list	*drive-channel-identifier* {*drive-channel-identifier*}
host-channel-identifier (four host ports per module)	(a1 | a2 | b1 | b2)
host-channel-identifier (eight host ports per module)	(a1 | a2 | a3 | a4 | b1 | b2 | b3 | b4)
host-channel-identifier (16 host ports per module)	(a1 | a2 | a3 | a4 | a5 | a6 | a7 | a8 | b1 | b2 | b3 | b4 | b5 | b6 | b7 | b8)
drive-type	(fibre | SATA | SAS)
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 | remoteMirror8 | remoteMirror16 | remoteMirror32 | remoteMirror64 | remoteMirror128 | volumeCopy | goldKey | mixedDriveTypes | highPerformanceTier | SSDSupport | safeStoreSecurity | safeStoreExternalKeyMgr | protectionInformation) 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*) [repositoryPoolUserLabel =*user-label*] [moduleLossProtect=(TRUE | FALSE)[1]]) | [drawerLossProtect=(TRUE | FALSE)[2]]) | (repositoryPool=*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 pool. Do not set a value for the moduleLossProtect parameter when you specify a pool.
count-based-repository- spec	repositoryRAIDLevel =*repository-raid-level* repositoryDriveCount=*integer-literal* [repositoryPoolUserLabel =*user-label*] [driveType=*drive-type*[4]] [moduleLossProtect=(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* |
	poolWidth=*integer-literal* |
	poolCount=*integer-literal* |
	volumesPerGroupCount=*integer-literal*[6] |
	hotSpareCount=*integer-literal* |
	segmentSize=*segment-size-spec* |
	cacheReadPrefetch=(TRUE | FALSE)
	securityType=(none | capable | enabled)[7] |
	protectionInformation=(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) |
	protectionInformation=(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*)

Recurring Syntax	Syntax Value
snapshot-schedule-attribute-value-list	*snapshot-schedule-attribute-value-pair* {*snapshot-schedule-attribute-value-pair*}

time-zone-spec	`(GMT+HH:MM | GMT-HH:MM)`
	`[dayLightSaving=HH:MM]`

snapshot-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 module loss protection to work, each drive in a pool must be in a separate module. If you set the `moduleLossProtect` parameter to TRUE and you have selected more than one drive from any one module, the storage array returns an error. If you set `moduleLossProtect` parameter to FALSE, the storage array performs operations, but the pool that you create might not have module loss protection.

If you set the `moduleLossProtect` parameter to TRUE, the storage array returns an error if the controller firmware cannot find drives that will enable the new pool to have module loss protection. If you set the `moduleLossProtect` parameter to FALSE, the storage array performs the operation even if it means that the pool might not have module loss protection.

[2] In modules 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 drawer loss protection to work, each drive in a pool must be in separate drawers. If you have a storage array configuration in which a pool spans several modules, you must make sure that the setting for drawer loss protection works with the setting for module loss protection. If you set the `moduleLossProtect` parameter to TRUE, you must set the `drawerLossProtect` parameter to TRUE. If you set the `trayLossProtect parameter` to TRUE, and you set the `drawerLossProtect parameter` to FALSE, 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 pool` 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 `poolWidth` 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 pool. If you want to set the `dataAssurance` parameter to `enabled`, all of the drives in the pool 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 pool.

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

[7] The `securityType` parameter enables you to specify the security setting for a pool 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 pool is not secure.
- `capable` – The pool is security capable, but security has not been enabled.
- `enabled` – The pool 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).

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
  - Pools
  - 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 back slashes (\\) 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.

When you enter a script command that requires a name, the script engine looks for a name that starts with an alphabetic character. The Script Engine might not recognize the following names:

- Names that are only numbers, such as 1 or 2
- Names that start with a number, such as 1Disk or 32Volume

To enter a name that consists only of numerical characters so that the Script Engine will recognize the name, use a combination of back slashes and double quotation marks. The following are examples of how you can enter names that consist only of numerical characters or start with numerical characters:

- `[\“1\”]`
- `[\“1Disk\”]`

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 module 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 modules and low-capacity drive modules. A high-capacity drive module has drawers that hold the drives. The drawers slide out of the drive module to provide access to the drives. A low-capacity drive module does not have drawers. For a high-capacity drive module, you must specify the identifier (ID) of the drive module, the ID of the drawer, and the ID of the slot in which a drive resides. For a low-capacity drive module, you need only specify the ID of the drive module and the ID of the slot in which a drive resides. For a low-capacity drive module, an alternative method for identifying a location for a drive is to specify the ID of the drive module, 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 module, 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 pool=2 userLabel=”v2” capacity=1gb owner=b;
create volume pool=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 pool=3 userLabel=”v5” capacity=1gb owner=a;
create volume pool=3 userLabel=”v6” capacity=1gb owner=b;
```

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.

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.

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.

Status Value	Meaning
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.
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.

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:

1. 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;
```

2. 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;
```

3. 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;
```

Firmware Compatibility Levels

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

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

NOTE The minimum controller firmware level indicates support by the software that releases the command, as well as support by all storage management software that picks up usage. CLI support capabilities depend on the hardware used. When an unsupported command is entered, an error message appears.

Examples of Firmware Compatibility Levels

The `create hostGroup` command has the following section.

Minimum Firmware Level

5.20

This level indicates that the entire script command runs under a minimum of controller firmware version 5.20.

The `show volume` command has the following section.

Minimum Firmware Level

5.00

5.43 adds the `summary` parameter

These notations indicate that the script command and all of the parameters except `summary` run under a minimum of controller firmware version 5.00. The `summary` parameter runs under a minimum of controller firmware version 5.43.

Chapter 2: Script Commands

ATTENTION The script commands are capable of damaging a configuration and causing loss of data access if not used correctly – Command operations are performed as soon as you run the commands. Some commands can immediately delete configurations or data. Before using the script commands, make sure that you have backed up all data, and have saved the current configuration so that you can reinstall it if the changes you make do not work.

The description of each scriptcommand is intended to provide all of the information that you need to be able to use the command. If, however, you have questions about command usage, these sections provide additional information that can help you use the script commands:

- "Naming Conventions" on page 15 lists the general rules for entering the names of storage array entities, such as volumes or drives, with the script commands.
- "Formatting CLI Commands" on page 15 lists the general formatting rules that apply to the CLI command wrapper.
- "Formatting Rules for Script Commands"on page 16 lists the general formatting rules that apply to the script command syntax.
- "Firmware Compatibility Levels" on page 20 explains how to interpret the firmware level information.
- "Commands Listed by Function" on page 21 lists the script commands organized into groups related to the physical features, the logical features, and the operational features of the storage array.
- "Commands Listed Alphabetically" on page 28 lists the script commands alphabetically and, for each script command, includes script command name, syntax, and parameters.

NOTE Terminology differences – The names of components and features change from time to time; however, the command syntax does not change at the same time. You will notice minor differences between the terminology used to describe components and features and the terminology used in the syntax to describe those same items when used in a command name, a parameter, or a variable.

Commands Listed by Function

Controller Commands

Clear Drive Channel Statistics

Diagnose Controller

Diagnose Controller iSCSI Host Cable

Enable Controller Data Transfer

Reset Controller

Save Controller NVSRAM

Save Drive Channel Fault Isolation Diagnostic Status

Set Controller

Set Controller Service Action Allowed Indicator

Set Drive Channel Status

Set Host Channel

Data Replicator Software Commands

Drive Commands

Host Topology Commands

iSCSI Commands

Module Commands

Pool Commands

Session Command

Snapshot Commands

Storage Array Commands

Re-create External Security Key

Reset Storage Array Battery Install Date

Reset Storage Array Diagnostic Data

Reset Storage Array Infiniband Statistics Baseline

Reset Storage Array iSCSI Baseline

Reset Storage Array RLS Baseline

Reset Storage Array SAS PHY Baseline

Reset Storage Array SOC Baseline

Reset Storage Array Volume Distribution

Save Storage Array Configuration

Save Storage Array DBM Database

Save Storage Array DBM Validator

Save Storage Array Diagnostic Data

Save Storage Array Events

Save Storage Array Firmware Inventory

Save Storage Array InfiniBand Statistics

Save Storage Array iSCSI Statistics

Save Storage Array Performance Statistics

Save Storage Array RLS Counts

Save Storage Array SAS PHY Counts

Save Storage Array SOC Counts

Save Storage Array State Capture

Save Storage Array Support Data

Set Storage Array

Set Storage Array ICMP Response

Set Storage Array iSNS Server IPv4 Address

Set Storage Array iSNS Server IPv6 Address

Set Storage Array iSNS Server Listening Port

Set Storage Array iSNS Server Refresh

Set Storage Array Learn Cycle

Set Storage Array Module Positions

Set Storage Array Redundancy Mode

Set Storage Array Remote Status Notification

Set Storage Array Security Key

Set Storage Array Time

Set Storage Array Unnamed Discovery Session

Set Storage Array Module Positions

Show Storage Array

Show Storage Array Auto Configure

Show Storage Array Host Topology

Show Storage Array LUN Mappings

Show Storage Array Negotiation Defaults

Show Storage Array Remote Status Notification

Show Storage Array Unconfigured iSCSI Initiators

Show Storage Array Unreadable Sectors

Start Secure Drive Erase

Start Storage Array iSNS Server Refresh

Start Storage Array Locate

Stop Storage Array Drive Firmware Download

Stop Storage Array iSCSI Session

Stop Storage Array Locate

Validate Storage Array Security Key

Uncategorized Commands

Show String

Volume Commands

Check Volume Parity

Clear Volume Reservations

Clear Volume Unreadable Sectors

Create RAID Volume (Automatic Drive Select)

Create RAID Volume (Free Extent Based Select)

Create RAID Volume (Manual Drive Select)

Delete Volume

Recover RAID Volume

Remove Volume LUN Mapping

Repair Volume Parity

Set Volume

Show Volume

Show Volume Action Progress

Show Volume Performance Statistics

Show Volume Reservations

Start Volume Initialization

Volume Copy Commands

Create Volume Copy

Recopy Volume Copy

Remove Volume Copy

Set Volume Copy

Show Volume Copy

Show Volume Copy Source Candidates

Show Volume Copy Target Candidates

Stop Volume Copy

Commands Listed Alphabetically

Activate Data Replicator Software Feature

This command creates the mirror repository volume and activates the Data Replicator Software premium feature. When you use this command, you can define the mirror repository volume in one of three ways:

- User-defined drives
- User-defined pool
- User-defined number of drives

If you choose to define a number of drives, the controller firmware chooses which drives to use for the mirror repository volume.

Syntax (User-Defined Drives)

```
activate storageArray feature=remoteMirror
repositoryRAIDLevel=(1 | 3 | 5 | 6)
repositoryDrives=(moduleID1,slotID1 ... moduleIDn,slotIDn)
repositoryPoolUserLabel=[poolName]
driveMediaType=(HDD | SSD | unknown | allMedia)
driveType=(fibre | SATA | SAS)
[moduleLossProtect=(TRUE | FALSE)
protectionInformation=(none | enabled)]
```

Syntax (User-Defined Pool)

```
activate storageArray feature=remoteMirror
repositoryPool=poolName
[freeCapacityArea=freeCapacityIndexNumber]
```

Syntax (User-Defined Number of Drives)

```
activate storageArray feature=remoteMirror
repositoryRAIDLevel=(1 | 3 | 5 | 6)
repositoryDriveCount=numberOfDrives
repositoryPoolUserLabel=[poolName]
driveMediaType=(HDD | SSD | unknown | allMedia)
driveType=(fibre | SATA | SAS)]
[moduleLossProtect=(TRUE | FALSE)
protectionInformation=(none | enabled)]
```

Activate Host Port

This command activates an inactive host port that was created when the Host Context Agent (HCA) registered the host port to a host.

Syntax

```
activate hostPort "userLabel"
```

Activate iSCSI Initiator

This command activates an inactive iSCSI initiator that was created when the Host Context Agent (HCA) registered the iSCSI initator to a host.

Syntax

```
activate iscsiInitiator "iscsiID"
```

Activate Storage Array Firmware

This command activates firmware that you have previously downloaded to the pending configuration area on the controllers in the storage array.

Syntax

```
activate storageArray firmware
```

Autoconfigure Storage Array

This command automatically configures a storage array. Before you enter the `autoConfigure storageArray` command, run the `show storageArray autoConfiguration` command. The `show storageArray autoConfiguration` command returns configuration information in the form of a list of valid drive types, RAID levels, volume information, and hot spare information. (This list corresponds to the parameters for the `autoConfigure storageArray` command.) The controllers audit the storage array and then determine the highest RAID level that the storage array can support and the most efficient volume definition for the RAID level. If the configuration that is described by the returned list is acceptable, you can enter the `autoConfigure storageArray` command without any parameters. If you want to modify the configuration, you can change the parameters to meet your configuration requirements. You can change a single parameter or all of the parameters. After you enter the `autoConfigure storageArray` command, the controllers set up the storage array by using either the default parameters or those you selected.

Syntax

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

Autoconfigure Storage Array Hot Spares

This command automatically defines and configures the hot spares in a storage array. You can run this command at any time. This command provides the best hot spare coverage for a storage array.

Syntax

```
autoConfigure storageArray hotSpares
```

Check Remote Mirror Status

This command returns the status of a remote-mirror volume. Use this command to determine when the status of the remote-mirror volume becomes Optimal.

Syntax

```
check remoteMirror localVolume [volumeName] optimalStatus timeout=timeoutValue
```

Check Volume Parity

This command checks a volume for parity and media errors and writes the results of the check to a file.

Syntax

```
check volume [volumeName]
parity [parityErrorFile=filename]
[mediaErrorFile=filename]
[priority=(highest | high | medium | low | lowest)]
[startingLBA=LBAvalue] [endingLBA=LBAvalue]
[verbose=(TRUE | FALSE)]
```

Clear Drive Channel Statistics

This command resets the statistics for all of the drive channels.

Syntax

```
clear all DriveChannels stats
```

Clear Storage Array Configuration

Use this command to perform one of these operations:

- Clear the entire storage array configuration, and return it back to the initial installation state
- Clear the configuration except for security information and identification information
- Clear pool configuration information and volume configuration information only

ATTENTION Possible damage to the storage array configuration – As soon as you run this command, the existing storage array configuration is deleted.

Syntax

```
clear storageArray configuration [all | pools]
```

Clear Storage Array Event Log

This command clears the Event Log in the storage array by deleting the data in the Event Log buffer.

ATTENTION Possible damage to the storage array configuration – As soon as you run this command, the existing Event Log in the storage array is deleted.

Syntax

```
clear storageArray eventLog
```

Clear Storage Array Firmware Pending Area

This command deletes a firmware image or NVSRAM values that you have previously downloaded from the pending area buffer.

ATTENTION Possible damage to the storage array configuration – As soon as you run this command, the contents of the existing pending area in the storage array are deleted.

Syntax

```
clear storageArray firmwarePendingArea
```

Clear Volume Reservations

This command clears persistent volume reservations.

Syntax

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

Clear Volume Unreadable Sectors

This command clears unreadable sector information from one or more volumes.

Syntax

```
clear (allVolumes | volume [volumeName] |
volumes [volumeName1 ... volumeNameN]) unreadableSectors
```

Create Host

This command creates a new host. If you do not specify a host group in which to create the new host, the new host is created in the Default Group.

Syntax

```
create host userLabel="hostName"
[hostGroup=("hostGroupName" | defaultGroup)]
[hostType=(hostTypeIndexLabel | hostTypeIndexNumber)]
```

Create Host Group

This command creates a new host group.

Syntax

```
create hostGroup userLabel="hostGroupName"
```

Create Host Port

This command creates a new host port identification on a host bus adapter (HBA) or on a host channel adapter (HCA). The identification is a software value that represents the physical HBA or HCA host port to the controller. Without the correct host port identification, the controller cannot receive instructions or data from the host port.

Syntax

```
create hostPort identifier=("wwID" | "gid")
userLabel="portLabel"
host="hostName"
interfaceType=(FC | SAS | IB)
```

Create iSCSI Initiator

This command creates a new iSCSI initiator object.

Syntax

```
create iscsiInitiator iscsiName="iscsiID"
userLabel="name"
host="hostName"
[chapSecret="securityKey"]
```

Create Pool

This command creates either a free-capacity pool or a pool with one volume when you enter a set of unassigned drives.

Syntax

```
create pool
drives=(moduleID1,slotID1 ... moduleIDn,slotIDn)
raidLevel=(0 | 1 | 3 | 5 | 6)
userLabel="poolName"
[driveMediaType=(HDD | SSD | unknown | allMedia)
driveType=(fibre | SATA | SAS)
moduleLossProtect=(TRUE | FALSE)
securityType=(none | capable | enabled)
protectionInformation=(none | enabled)]
```

Create RAID Volume (Automatic Drive Select)

This command creates a pool across the drives in the storage array and a new volume in the pool. The storage array controllers choose the drives to be included in the volume.

NOTE If you have drives with different capacities, you cannot automatically create volumes by specifying the `driveCount` parameter. If you want to create volumes with drives of different capacities, see "Create RAID Volume (Manual Drive Select)."

Syntax

```
create volume driveCount=numberOfDrives
poolUserLabel="poolName"
raidLevel=(0 | 1 | 3 | 5 | 6)
userLabel="volumeName"
driveMediaType=(HDD | SSD | unknown | allMedia)
[driveType=(fibre | SATA | SAS)
capacity=volumeCapacity
owner=(a | b)
cacheReadPrefetch=(TRUE | FALSE)
segmentSize=segmentSizeValue
usageHint=(fileSystem | dataBase | multiMedia)
moduleLossProtect=(TRUE | FALSE)
dssPreAllocate=(TRUE | FALSE)
securityType=(none | capable | enabled)
protectionInformation=(none | enabled)]
```

Create RAID Volume (Free Extent Based Select)

This command creates a volume in the free space of a pool.

Syntax

```
create volume pool="poolName"
userLabel="volumeName"
[freeCapacityArea=freeCapacityIndexNumber
capacity=volumeCapacity
owner=(a | b)
cacheReadPrefetch=(TRUE | FALSE)
segmentSize=segmentSizeValue
usageHint=(fileSystem | dataBase | multiMedia)]
[dssPreAllocate=(TRUE | FALSE)
securityType=(none | capable | enabled)
protectionInformation=(none | enabled)]
```

Create RAID Volume (Manual Drive Select)

This command creates a new pool and volume and lets you specify the drives for the volume.

NOTE You cannot use mixed drive types in the same pool and volume. This command fails if you specify different types of drives for the RAID volume.

Syntax

```
create volume drives=(moduleID1,slotID1 ... moduleIDn,slotIDn)
poolUserLabel="poolName"
raidLevel=(0 | 1 | 3 | 5 | 6)
userLabel="volumeName"
[capacity=volumeCapacity
owner=(a | b)
cacheReadPrefetch=(TRUE | FALSE)
segmentSize=segmentSizeValue
usageHint=(fileSystem | dataBase | multiMedia)
moduleLossProtect=(TRUE | FALSE)
dssPreAllocate=(TRUE | FALSE)
securityType=(none | capable | enabled)
protectionInformation=(none | enabled)]
```

Create Remote Mirror

This command creates both the primary volume and the secondary volume for a remote-mirror pair. This command also sets the write mode (synchronous write mode or asynchronous write mode) and the synchronization priority.

Syntax

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

Create Snapshot Volume

This command creates a snapshot volume of a base volume. You can also use this command to create a new repository pool if one does not already exist, or if you would prefer a different repository pool. This command defines three ways to create a snapshot volume:

- In a new repository pool created from user-defined drives
- In a new repository pool created from a user-defined number of drives
- In an existing repository pool

If you choose to define a number of drives, the controller firmware chooses which drives to use for the snapshot volume.

Syntax (User-Defined Drives)

```
create snapshotVolume baseVolume="baseVolumeName"
(repositoryRAIDLevel=(1 | 3 | 5 | 6)
repositoryDrives=(moduleID1,slotID1 ... moduleIDn,slotIDn))
[repositoryPoolUserLabel="repositoryPoolName"
moduleLossProtect=(TRUE | FALSE)
freeCapacityArea=freeCapacityIndexNumber
userLabel="snapshotVolumeName"
warningThresholdPercent=percentValue
repositoryPercentOfBase=percentValue
repositoryUserLabel="repositoryName"
repositoryFullPolicy=(failBaseWrites | failSnapshot) |
enableSchedule=(TRUE | FALSE) |
schedule=(immediate | snapshotSchedule)]
```

Syntax (User-Defined Number of Drives)

```
create snapshotVolume baseVolume="baseVolumeName"
repositoryRAIDLevel=(1 | 3 | 5 | 6)
repositoryDriveCount=numberOfDrives
[repositoryPoolUserLabel="repositoryPoolName"
driveMediaType=(HDD | SSD | unknown | allMedia)]
driveType=(fibre | SATA | SAS)
moduleLossProtect=(TRUE | FALSE)
userLabel="snapshotVolumeName"
warningThresholdPercent=percentValue
repositoryPercentOfBase=percentValue
repositoryUserLabel="repositoryName"
repositoryFullPolicy=(failBaseWrites | failSnapshot) | enableSchedule=(TRUE |
FALSE) |
schedule=(immediate | snapshotSchedule)]
```

Syntax (Existing Repository Pool)

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

Create Storage Array Security Key

This command creates a new security key for a storage array that has Encryption Services (FDE) drives. This command also sets the security definitions and sets the state to Security Enabled.

NOTE Before you create a storage array security key, you must set the password for the storage array. Use the `set storageArray` command to set the password for the storage array.

Syntax

```
create storageArray securityKey
[keyIdentifier="keyIdentifierString"] |
passPhrase="passPhraseString" |
file="fileName" |
commitSecurityKey=(TRUE | FALSE)
```

Create Volume Copy

This command creates a volume copy and starts the volume copy operation.

ATTENTION Starting a volume copy operation overwrites all existing data on the target volume, makes the target volume read-only to hosts, and fails all snapshot volumes associated with the target volume, if any exist. If you have used the target volume as a copy before, be sure you no longer need the data or have it backed up.

This command creates volume copies in two ways:

- Volume copy without snapshot
- Volume copy with snapshot

If you use volume copy without snapshot you cannot write to the source volume until the copy operation is complete. If you want to be able to write to the source volume before the copy operation is complete, use volume copy with snapshot. You can select volume copy with snapshot through the optional parameters in the command syntax.

After completion of the volume copy with snapshot operation, the snapshot is disabled.

NOTE You can have a maximum of eight volume copies in progress at one time. If you try to create more than eight volume copies at one time, the controllers return a status of Pending until one of the volume copies that is in progress finishes and returns a status of Complete.

Syntax

```
create volumeCopy source="sourceName"
target="targetName"
[copyPriority=(highest | high | medium | low | lowest)
targetReadOnlyEnabled=(TRUE | FALSE)
copyType=(offline | online)
repositoryPercentOfBase=(20 |40 | 60 | 120 | default) |
repositoryGroupPreference=(sameAsSource | otherThanSource |
... default)]
```

Deactivate Remote Mirror

This command deactivates the Data Replicator Software premium feature, disassembles the mirror repository volume, and releases the controller owner of the secondary volume. The controller host port that is dedicated to the secondary volume is available for host data transfers.

Syntax

```
deactivate storageArray feature=remoteMirror
```

Delete Host

This command deletes one or more hosts.

Syntax

```
delete (host [hostName] |
hosts ["hostName1" ... "hostNameN"])
```

Delete Host Group

This command deletes a host group.

ATTENTION Possible damage to the storage array configuration – This command deletes all of the host definitions in the host group.

Syntax

```
delete hostGroup [hostGroupName]
```

Delete Host Port

This command deletes a host port identification. The identification is a software value that represents the physical host port to the controller. By deleting the identification, the controller no longer recognizes instructions and data from the host port.

Syntax

```
delete hostPort [hostPortName]
```

Delete iSCSI Initiator

This command deletes a specific iSCSI initiator object.

Syntax

```
delete iscsiInitiator (["iscsiID"] | ["name"])
```

Delete Pool

ATTENTION Possible damage to the storage array configuration – All of the data in the pool is lost as soon as you run this command.

This command deletes an entire pool and its associated volumes.

Syntax

```
delete pool [poolName]
```

Delete Snapshot Volume

This command deletes one or more snapshot volumes or snapshot repository volumes. You can also use this command to remove schedules for creating snapshots.

ATTENTION Possible damage to the storage array configuration – All of the data in the volume is lost as soon as you run this command.

Syntax

```
delete snapshot (volume [volumeName] |
volumes [volumeName1 ... volumeNameN])
[schedule]
```

Delete Volume

This command deletes one or more standard volumes, snapshot volumes, or snapshot repository volumes.

ATTENTION Possible damage to the storage array configuration – All of the data in the volume is lost as soon as you run this command.

Syntax

```
delete (allVolumes |
volume [volumeName] |
volumes [volumeName1 ... volumeNameN])
removePool=(TRUE | FALSE)
```

Diagnose Controller

This command runs diagnostic tests on the controller. The diagnostic tests consist of loopback tests in which data is written to the drives and read from the drives.

Syntax

```
diagnose controller [(a | b)]
loopbackDriveChannel=(allchannels | (1 | 2 | 3 | 4 | 5 | 6 | 7 | 8))
testID=(1 | 2 | 3 | discreteLines)
[patternFile="filename"]
```

Diagnose Controller iSCSI Host Cable

This command runs diagnostic tests on the copper cables between iSCSI Host interface cards and a controller. You can run diagnostics on a selected port or all ports. The ports must be able to support the cable diagnostics. If the ports do not support cable diagnostics an error is returned.

Syntax

```
diagnose controller [(a | b)]
iscsiHostPorts=(all | ("wwID" | "gID")
testID=cableDiagnostics
```

Diagnose Remote Mirror

This command tests the connection between the specified primary volumes and the mirror volumes on a storage array with the Data Replicator Software premium feature enabled.

Syntax

```
diagnose remoteMirror (primary [primaryVolumeName] |
primaries [primaryVolumeName1 ... primaryVolumeNameN])
testID=connectivity
```

Disable External Security Key Management

This command disables external security key management for a storage array that has Encryption Services drives.

Syntax

```
disable storageArray externalKeyManagement
file="fileName"
passPhrase="passPhraseString"
```

Disable Storage Array Feature

This command disables a storage array premium feature. Run the `show storageArray` command to show a list of the feature identifiers for all enabled premium features in the storage array.

Syntax

```
disable storageArray [featurePack |
feature=(storagePartition2 | storagePartition4 |
storagePartition8 | storagePartition16 | storagePartition32 |
storagePartition64 | storagePartition96 | storagePartition128 |
storagePartition256 | storagePartitionMax |
snapshot2 | snapshot4 | snapshot8 | snapshot16 |
remoteMirror8 | remoteMirror16 | remoteMirror32 |
remoteMirror64 | remoteMirror128 | volumeCopy | goldKey |
mixedDriveTypes | highPerformanceTier | SSDSupport |
safeStoreSecurity | safeStoreExternalKeyMgr)]
```

Disable Storage Array Remote Status Notification

This command turns off the remote status notification feature. The remote status notification feature enables the periodic collection of the storage array profile and the support bundle information by the persistent monitor. The storage array profile and the support bundle information are automatically sent to a support data collection web server. To turn on the remote status notification feature, use the `enable storageArray remoteStatusNotification` command.

Syntax

```
disable storageArray remoteStatusNotification
```

Download Drive Firmware

This command downloads a firmware image to a drive.

ATTENTION Possible damage to the storage array configuration – Downloading drive firmware incorrectly can result in damage to the drives or a loss of data access.

This command is intended for downloading a firmware image to only one drive at a time. If you use this command in a script, make sure that you use this command only once. If you use this command more than once, the operation can fail. You can download firmware images to all of the drives in a storage array at one time by using the `download storageArray driveFirmware` command.

Syntax

```
download drive [moduleID,slotID] firmware file="filename"
```

Download Environmental Card Firmware

This command downloads environmental services monitor (ESM) firmware.

Syntax

```
download (allModules | module [moduleID])
firmware file="filename"
```

Download Module Configuration Settings

This command downloads the factory default settings to all of the drive modules in a storage array or to a specific drive module in a storage array.

Syntax

```
download (allModules | module [moduleID]) configurationSettings
firmware file="filename"
```

Download Power Supply Firmware

This command downloads firmware updates to the power supplies. You can schedule simultaneous firmware updates for several power supplies, and the power supplies can be in different modules. A single firmware file can contain updates for several different power supplies. Matching firmware updates are automatically chosen for the power supplies. Firmware download occurs only if the new firmware version is not the same as the version of the power supplies on the modules. A download succeeds only if the power supply is in an Optimal state and there is a redundant power supply that is in an Optimal state.

To bypass these checks 'forceUpdate' can be used.

Syntax

```
download (allModules |
...module [moduleID1]...[moduleIDn] |
...module [moduleID])
powerSupplyUpdate file="filename"
powerSupplyUnit [(left | right) | (top | bottom)] |
[forceUpdate]
```

Download Storage Array Drive Firmware

This command downloads firmware images to all of the drives in the storage array.

Syntax

```
download storageArray driveFirmware file="filename"
[file="filename2"... file="filenameN"]
```

Download Storage Array Firmware/NVSRAM

This command downloads firmware and, optionally, NVSRAM values for the storage array controller. If you want to download only NVSRAM values, use the `downLoad storageArray NVSRAM` command.

Syntax

```
download storageArray firmware [, NVSRAM ]
file="filename" [, "NVSRAM-filename"]
[downgrade=(TRUE | FALSE)]
[activateNow=(TRUE | FALSE)]
```

Download Storage Array NVSRAM

This command downloads the NVSRAM values for the storage array controller.

Syntax

```
download storageArray NVSRAM file="filename"
```

Enable Controller Data Transfer

This command revives a controller that has become quiesced while running diagnostics.

Syntax

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

Enable External Security Key Management

This command enables external security key management for a storage array that has Encryption Services drives.

Syntax

```
enable storageArray externalKeyManagement
file="fileName" |
passPhrase="passPhraseString"
```

Enable Pool Security

This command converts a non-secure pool to a secure pool.

Syntax

```
enable pool [poolName] security
```

Enable Storage Array Feature

This command enables a premium feature by using a feature key file.

ATTENTION Before you enable the High Performance Tier premium feature, stop all host I/O operations to the storage array. When you enable the High Performance Tier premium feature, both controllers in the storage array will immediately reboot.

Syntax

```
enable storageArray [featurePack | feature]
file="filename"
```

Enable Storage Array Remote Status Notification

This command turns on the remote status notification feature. The remote status notification feature enables the periodic collection of the storage array profile and the support bundle information by the persistent monitor. The storage array profile and the support bundle information are automatically sent to a support data collection web server. To turn off the remote status notification feature, use the `disable storageArray remoteStatusNotification` command.

Syntax

```
enable storageArray remoteStatusNotification
```

Export Storage Array Security Key

This command saves a Encryption Services (ES) security key to a file. You can transfer the file from one storage array to another storage array. The file enables you to move ES drives between storage arrays.

Syntax

```
export storageArray securityKey
passPhrase="passPhraseString"
file="fileName"
```

Import Storage Array Security Key

This command unlocks one or more Encryption Services (ES) drives that you have imported from one storage array to another storage array. Only the ES drives with the matching security key from the imported storage array are unlocked. After they are unlocked, the security key for the new storage array is applied.

Syntax

```
import storageArray securityKey file="fileName"
passPhrase="passPhraseString"
```

Load Storage Array DBM Database

This command uploads a Database Management (DBM) database image from a file. This command restores a storage array to the exact configuration that existed when the DBM database image was captured to a file using the `save storageArray dbmDatabase` command. Before using this command, you must first obtain a validator or a security code from your Sun Customer Care Center representative. To obtain a validator, use the `save storageArray dbmValidator` command to generate an XML file that contains validator information. Your Sun Customer Care Center representative uses the XML file to generate the validator required for this command.

Syntax

```
load storageArray dbmDatabase
file="filename" validator=validatorValue
```

Recopy Volume Copy

This command reinitiates a volume copy operation using an existing volume copy pair.

ATTENTION Starting a volume copy operation overwrites all existing data on the target volume, makes the target volume read-only to hosts, and fails all snapshot volumes associated with the target volume, if any exist. If you have used the target volume as a copy before, be sure you no longer need the data or have it backed up.

This command works with volume copy pairs that you created with a snapshot volume or without a snapshot volume.

Syntax

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

Recover RAID Volume

This command creates a RAID volume with the given properties without initializing any of the user data areas on the drives. Parameter values are derived from the Recovery Profile data file (`recoveryProfile.csv`) for the storage array. You can create the recover volume in an existing pool or create a new pool by using this command.

NOTE You can run this command only from a command line. You cannot run this command from the GUI script editor. You cannot use the storage managment GUI to recover a volume.

Syntax

```
recover volume (drive=(moduleID,slotID) |
drives=(moduleID1,slotID1 ... moduleIDn,slotIDn) |
pool=poolName))
[newPool=poolName]
userLabel=("volumeName"
volumeWWN="volumeWWN")
capacity=volumeCapacity
offset=offsetValue
raidLevel=(0 | 1 | 3 | 5 | 6)
segmentSize=segmentSizeValue
dssPreallocate=(TRUE | FALSE)
SSID=volumeCapacity
owner=(a | b)
cacheReadPrefetch=(TRUE | FALSE)
protectionInformation=(none | enabled)]
```

Re-create External Security Key

This command regenerates a storage array security key for use with the external security key management feature.

Syntax

```
recreate storageArray securityKey
passPhrase="passPhraseString"
file="fileName"
```

Re-create Data Replicator Software Repository Volume

This command creates a new Data Replicator Software repository volume (also called a mirror repository volume) by using the parameters defined for a previous mirror repository volume. The underlying requirement is that you have previously created a mirror repository volume. When you use this command, you can define the mirror repository volume in one of three ways: user-defined drives, user-defined pool, or user-defined number of drives for the mirror repository volume. If you choose to define a number of drives, the controller firmware chooses which drives to use for the mirror repository volume.

Syntax (User-Defined Drives)

```
recreate storageArray mirrorRepository
repositoryRAIDLevel=(1 | 3 | 5 | 6)
repositoryDrives=(moduleID1,slotID1 ... moduleIDn,slotIDn)
[moduleLossProtect=(TRUE | FALSE)
protectionInformation=(none | enabled)]
```

Syntax (User-Defined Pool)

```
recreate storageArray mirrorRepository
repositoryPool=poolName [freeCapacityArea=freeCapacityIndexNumber]
```

Syntax (User-Defined Number of Drives)

```
recreate storageArray mirrorRepository
repositoryRAIDLevel=(1 | 3 | 5 | 6)
repositoryDriveCount=numberOfDrives
[driveType=(fibre | SATA | SAS)]
[moduleLossProtect=(TRUE | FALSE)
protectionInformation=(none | enabled)]
```

Re-create Snapshot

This command starts a fresh copy-on-write operation by using an existing snapshot volume. You can re-create a single snapshot volume or re-create multiple snapshot volumes. If you choose to re-create multiple snapshot volumes, you can re-create from two to the maximum number of snapshot volumes that your storage array can support.

Syntax

```
recreate snapshot (volume [volumeName] |
volumes [volumeName1 ... volumeNameN])
[userLabel="snapshotVolumeName"
warningThresholdPercent=percentValue
repositoryFullPolicy (failBaseWrites | failSnapshot)]
```

Remove Remote Mirror

This command removes the mirror relationship between the primary volume and the secondary volume in a remote-mirror pair.

Syntax

```
remove remoteMirror (localVolume [volumeName] |
localVolumes [volumeName1 ... volumeNameN])
```

Remove Volume Copy

This command removes a volume copy pair.

Syntax

```
remove volumeCopy target [targetName]
[source [sourceName]
copyType=(online | offline)]
```

Remove Volume LUN Mapping

This command removes the logical unit number (LUN) mapping from one or more volumes.

Syntax

```
remove (allVolumes | volume [volumeName] |
volumes [volumeName1 ... volumeNameN] | accessVolume)
lunMapping (host="hostName" |
hostGroup=("hostGroupName" | defaultGroup))
```

Repair Volume Parity

This command repairs the parity errors on a volume.

Syntax

```
repair volume [volumeName] parity
parityErrorFile="filename"
[verbose=(TRUE | FALSE)]
```

Replace Drive

This command redefines the composition of a pool. You can use this command to replace a drive with either an unassigned drive or a fully integrated hot spare.

Syntax

```
replace drive([moduleID,slotID] | <"wwID">)
replacementDrive=moduleID,slotID
```

Reset Controller

This command resets a controller, and it is disruptive to I/O operations.

ATTENTION When you reset a controller, the controller is removed from the data path and is not available for I/O operations until the reset operation is complete. If a host is using volumes that are owned by the controller being reset, the I/O 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 use these volumes.

Syntax

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

Notes

The controller that receives the reset controller command resets the controller specified. For example, if the reset controller command is sent to controller A to request a reset of controller A, then controller A reboots itself by doing a soft reboot. If the reset controller command is sent to controller A to request a reset of controller B, then controller A holds controller B in reset and then releases controller B from reset, which is a hard reboot. A soft reboot in some products only resets the IOC chip. A hard reboot resets both the IOC and the expander chips in the controller.

Reset Storage Array Battery Install Date

This command resets the age of the batteries in a storage array to zero days. You can reset the age of the batteries for an entire storage array or the age of a battery in a specific controller or in a specific battery pack.

Syntax

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

Reset Storage Array Diagnostic Data

This command resets the NVSRAM that contains the diagnostic data for the storage array. This command does not delete the diagnostic data. This command replaces the Needs Attention status with the Diagnostic Data Available status. The old diagnostic data is written over automatically when new data is captured. The memory that contains the diagnostic data is also cleared when the controllers reboot. Before you reset the diagnostic data, use the `save storageArray diagnosticData` command to save the diagnostic data to a file.

ATTENTION Run this command only with the assistance of your Sun Customer Care Center representative.

Syntax

```
reset storageArray diagnosticData
```

Reset Storage Array Infiniband Statistics Baseline

This command resets the Infiniband statistics baseline to 0 for the storage array.

Syntax

```
reset storageArray ibStatsBaseline
```

Reset Storage Array iSCSI Baseline

This command resets the iSCSI baseline to 0 for the storage array.

Syntax

```
reset storageArray iscsiStatsBaseline
```

Reset Storage Array RLS Baseline

This command resets the read link status (RLS) baseline for all devices by setting all of the RLS counts to 0.

Syntax

```
reset storageArray RLSBaseline
```

Reset Storage Array SAS PHY Baseline

This command resets the SAS physical layer (SAS PHY) baseline for all devices execpt the drives, and removes the list of errors from the `.csv` file. The `.csv` file is generated when you run the `save storageArray SASPHYCounts` command.

NOTE The `reset storageArray SASPHYBaseline` command clears error counts for all devices except the drives. After you run this command, the `.csv` file will continue to list the `Drive`PHY errors. All other errors are deleted from the `.csv` file.

Syntax

```
reset storageArray SASPHYBaseline
```

Reset Storage Array SOC Baseline

This command resets the baseline for all switch-on-a-chip (SOC) devices that are accessed through the controllers. This command resets the baseline by setting all of the SOC counts to 0. This command is valid only for Fibre Channel devices in an arbitrated loop topology.

Syntax

```
reset storageArray SOCBaseline
```

Reset Storage Array Volume Distribution

This command reassigns (moves) all of the volumes to their preferred controller.

Syntax

```
reset storageArray volumeDistribution
```

Resume Remote Mirror

This command resumes a suspended Data Replicator Software operation.

Syntax

```
resume remoteMirror (primary [volumeName] |
primaries [volumeName1 ... volumeNameN])
[writeConsistency=(TRUE | FALSE)}
```

Revive Drive

This command forces the specified drive to the Optimal state.

ATTENTION Possible loss of data access – Correct use of this command depends on the data configuration on all of the drives in the pool. Never try to revive a drive unless you are supervised by your Sun Customer Care Center representative.

Syntax

```
revive drive [moduleID,slotID]
```

Revive Pool

This command forces the specified pool and its associated failed drives to the Optimal state.

ATTENTION Possible loss of data access – Correct use of this command depends on the data configuration on all of the drives in the pool. Never try to revive a drive unless you are supervised by your Sun Customer Care Center representative.

Syntax

```
revive pool [poolName]
```

Save Controller NVSRAM

This command saves a copy of the controller NVSRAM values to a file. This command saves all of the regions.

Syntax

```
save controller [(a | b)] NVSRAM file="filename"
```

Save Drive Channel Fault Isolation Diagnostic Status

This command saves the drive channel fault isolation diagnostic data that is returned from the `start driveChannel faultDiagnostics` command. You can save the diagnostic data to a file as standard text or as XML.

See "Start Drive Channel Fault Isolation Diagnostics" for more information.

Syntax

```
save driveChannel faultDiagnostics file="filename"
```

Save Drive Log

This command saves the log sense data to a file. Log sense data is maintained by the storage array for each drive.

Syntax

```
save allDrives logFile="filename"
```

Save Storage Array Configuration

This command creates a script file that you can use to re-create the current storage array volume configuration.

Syntax

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

Save Storage Array DBM Database

This command saves the current state of the storage array's Database Management (DBM) database into a local file. The output file that is produced can be used as the input file for the `save storageArray dbmValidator` and the `load storageArray dbmDatabase` commands.

Syntax

```
save storageArray dbmDatabase file="filename"
```

Save Storage Array DBM Validator

This command saves a storage array's Database Management (DBM) validation information in an XML file, which can be used by a Sun Customer Care Center representative to generate a security code or Validator. The Validator must be included in the `load storageArray dbmDatabase` command when restoring a storage array back to a pre-existing configuration.

Syntax

```
save storageArray dbmValidatorInfo file="filename" dbmDatabase="filename"
```

Save Storage Array Diagnostic Data

This command saves the storage array diagnostic data from either the controllers or the environmental services monitors (ESMs) to a file. You can review the file contents at a later time. You can also send the file to your Sun Customer Care Center representative for further review.

After you have saved the diagnostic data, you can reset the NVSRAM registers that contain the diagnostic data so that the old data can be overwritten. Use the `reset storageArray diagnosticData` command to reset the diagnostic data registers.

ATTENTION Run this command only with the assistance of your Sun Customer Care Center representative.

Syntax

```
save storageArray diagnosticData [(controller | esm)]
file="filename"
```

Save Storage Array Events

This command saves events from the Major Event Log to a file. You can save these events:

- **Critical events** – An error occurred on the storage array that needs to be addressed immediately. Loss of data access might occur if you do not immediately correct the error.
- **Warning events** – An error occurred on the storage array that results in degraded performance or reduced ability to recover from another error. Access to data has not been lost, but you must correct the error to prevent possible loss of data access if another error would occur.
- **Informational events** – An event occurred on the storage array that does not impact normal operations. The event is reporting a change in configuration or other information that might be useful in evaluating how well the storage array is performing.
- **Debug events** – An event occurred on the storage array that provides information that you can use to help determine the steps or states that led to an error. You can send a file with this information to your Sun Customer Care Center representative to help determine the cause of an error.

NOTE Some storage arrays might not be able to support all four types of events.

Syntax

```
save storageArray (allEvents | criticalEvents |
warningEvents | infoEvents | debugEvents)
file="filename"
[count=numberOfEvents
forceSave=(TRUE | FALSE)]
```

Save Storage Array Firmware Inventory

This command saves a report to a file of all of the firmware currently running on the storage array. The report lists the firmware for these components:

- Controllers
- Drives
- Environmental services monitors (ESMs)

You can use the information to help identify out-of-date firmware or firmware that does not match the other firmware in your storage array. You can also send the report to your Sun Customer Care Center representative for further review.

Syntax

```
save storageArray firmwareInventory file="filename"
```

Save Storage Array InfiniBand Statistics

This command saves the InfiniBand performance statistics of the storage array to a file.

Syntax

```
save storageArray ibStats  [raw | baseline]
file="filename"
```

Save Storage Array iSCSI Statistics

This command saves the iSCSI performance of the storage array to a file.

Syntax

```
save storageArray iscsiStatistics [raw | baseline] file="filename"
```

Save Storage Array Performance Statistics

This command saves the performance statistics to a file. Before you use this command, run the `set session performanceMonitorInterval` command and the `set session performanceMonitorIterations` command to specify how often statistics are collected.

Syntax

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

Save Storage Array RLS Counts

This command saves the read link status (RLS) counters to a file.

Syntax

```
save storageArray RLSCounts file="filename"
```

Save Storage Array SAS PHY Counts

This command saves the SAS physical layer (SAS PHY) counters to a file. To reset the SAS PHY counters, run the `reset storageArray SASPHYBaseline` command.

Syntax

```
save storageArray SASPHYCounts file="filename"
```

Save Storage Array SOC Counts

This command saves the SOC error statistics to a file. This command is valid only for Fibre Channel devices in an arbitrated loop topology.

Syntax

```
save storageArray SOCCounts file="filename"
```

Save Storage Array State Capture

This command saves the state capture of a storage array to a file.

Syntax

```
save storageArray stateCapture file="filename"
```

Save Storage Array Support Data

This command saves the support-related information of the storage array to a file. Support-related information includes these items:

- The storage array profile
- The Major Event Log information
- The read link status (RLS) data
- The NVSRAM data
- Current problems and associated recovery information
- The performance statistics for the entire storage array
- The persistent registration information and the persistent reservation information
- Detailed information about the current status of the storage array
- The diagnostic data for the drive
- A recovery profile for the storage array
- The unreadable sectors that are detected on the storage array
- The state capture data
- An inventory of the versions of the firmware running on the controllers, the drives, and the environmental services monitors (ESMs)

Syntax

```
save storageArray supportData file="filename"
```

Save Module Log

This command saves the log sense data to a file. Log sense data is maintained by the environmental cards for each module. Not all of the environmental cards contain log sense data.

Syntax

```
save allModules logFile="filename"
```

Set Controller

This command defines the attributes for the controllers.

Syntax

```
set controller [(a | b)]
availability=(online | offline | serviceMode) |
ethernetPort [(1 | 2)] ethernetPortOptions |
globalNVSRAMByte [nvsramOffset]=(nvsramByteSetting | nvsramBitSetting) |
hostNVSRAMByte [hostType, nvsramOffset]=(nvsramByteSetting | nvsramBitSetting) |
IPv4GatewayIP=ipAddress |
IPv6RouterAddress=ipv6Address |
iscsiHostPort [(1 | 2 | 3 | 4)] iscsiHostPortOptions
rloginEnabled=(TRUE | FALSE) |
serviceAllowedIndicator=(on | off)
```

Set Controller Service Action Allowed Indicator

This command turns on or turns off the Service Action Allowed indicator light on a controller in a controller module or an array module. If the storage array does not support the Service Action Allowed indicator light feature, this command returns an error. If the storage array supports the command but is unable to turn on or turn off the indicator light, this command returns an error. (To turn on or turn off the Service Action Allowed indicator light on the power-fan CRU or the interconnect-battery CRU, use the `set module serviceAllowedIndicator` command.)

Syntax

```
set controller=[(a | b)]
serviceAllowedIndicator=(on | off)
```

Set Drive Channel Status

This command defines how the drive channel performs.

Syntax

```
set driveChannel [(1 | 2 | 3 | 4 | 5 | 6 | 7 | 8)]
status=(optimal | degraded)
```

Set Drive Hot Spare

This command assigns or unassigns one or more drives as a hot spare.

Syntax

```
set (drive [moduleID,slotID] |
drives [moduleID1,slotID1 ... moduleIDn,slotIDn])
hotSpare=(TRUE | FALSE)
```

Set Drive Service Action Allowed Indicator

This command turns on or turns off the Service Action Allowed indicator light on a drive in drive modules that support the Service Action Allowed indicator light feature. If the storage array does not support the Service Action Allowed indicator light feature, this command returns an error. If the storage array supports the command but is unable to turn on or turn off the indicator light, this command returns an error.

Syntax

```
set (drive [moduleID,slotID] |
drives [moduleID1,slotID1 ... moduleIDn,slotIDn])
serviceAllowedIndicator=(on | off)
```

Set Drive State

This command sets a drive to the Failed state. (To return a drive to the Optimal state, use the `revive drive` command.)

Syntax

```
set (drive [moduleID,slotID] |
operationalState=failed
set (drive [moduleID,slotID] |
operationalState=failed
```

Set Foreign Drive to Native

A drive is considered to be native when it is a part of a pool in a storage array. A drive is considered to be foreign when it does not belong to a pool in a storage array or when it fails to be imported with the drives of a pool that are transferred to a new storage array. The latter failure creates an incomplete pool on the new storage array.

Run this command to add the missing (foreign) drives back into their original pool and to make them part of the poolin the new storage array.

Use this operation for emergency recovery only: when one or more drives need to be changed from a foreign drive status and returned to a native status within their original pool.

ATTENTION Possible data corruption or data loss – Using this command for reasons other than what is stated previously might result in data loss without notification.

Syntax

```
set (drive [moduleID,slotID] |
drives [moduleID1,slotID1 ... moduleIDn,slotIDn] | allDrives) nativeState
```

Set Host

This command assigns a host to a host group or moves a host to a different host group. You can also create a new host group and assign the host to the new host group with this command. The actions performed by this command depend on whether the host has individual mappings or does not have individual mappings.

Syntax

```
set host [hostName]
hostGroup=("hostGroupName" | none | defaultGroup)
userLabel="newHostName"
hostType=(hostTypeIndexLabel | hostTypeIndexNumber)
```

Set Host Channel

This command defines the loop ID for the host channel.

Syntax

```
set hostChannel [hostChannelNumber]
preferredID=portID
```

Set Host Group

This command renames a host group.

Syntax

```
set hostGroup [hostGroupName]
userLabel="newHostGroupName"
```

Set Host Port

This command changes the host type for a host port. You can also change a host port label with this command.

Syntax

```
set hostPort [portLabel] host="hostName" userLabel="newPortLabel"
```

Set iSCSI Initiator

This command sets the attributes for an iSCSI initiator.

Syntax

```
set iscsiInitiator (["iscsiID"] |
userLabel="newName" |
host="newHostName" |
chapSecret="newSecurityKey")
```

Set iSCSI Target Properties

This command defines properties for an iSCSI target.

Syntax

```
set iscsiTarget ["userLabel"]
authenticationMethod=(none | chap) |
chapSecret=securityKey |
targetAlias="userLabel"
```

Set Remote Mirror

This command defines the properties for a remote-mirror pair.

Syntax

```
set remoteMirror (localVolume [volumeName] |
localVolumes [volumeName1 ... volumeNameN])
role=(primary | secondary)
[force=(TRUE | FALSE)]
syncPriority=(highest | high | medium | low | lowest)
autoResync=(enabled | disabled)
writeOrder=(preserved | notPreserved)
writeMode=(synchronous | asynchronous)
```

Set Session

This command defines how you want the current script engine session to run.

Syntax

```
set session errorAction=(stop | continue)
password="storageArrayPassword"
performanceMonitorInterval=intervalValue
performanceMonitorIterations=iterationValue
```

Set Snapshot Volume

This command defines the properties for a snapshot volume and lets you rename a snapshot volume.

Syntax

```
set (volume [volumeName] |
volumes [volumeName1 ... volumeNameN])
userLabel="snapshotVolumeName"
warningThresholdPercent=percentValue
repositoryFullPolicy=(failBaseWrites | failSnapshot) |
enableSchedule=(TRUE | FALSE) |
schedule=(immediate | snapshotSchedule)
```

Set Storage Array

This command defines the properties of the storage array.

Syntax

```
set storageArray {alarm=(enable | disable | mute) |
{autoSupportConfig (enable | disable) |
cacheBlockSize=cacheBlockSizeValue |
cacheFlushStart=cacheFlushStartSize |
cacheFlushStop=cacheFlushStopSize |
defaultHostType=("hostTypeName" | hostTypeIdentifier)
failoverAlertDelay=delayValue |
mediaScanRate=(disabled | 1-30) |
password="password" |
userLabel="storageArrayName"
isnsRegistration=(TRUE | FALSE))
```

Set Storage Array ICMP Response

This command returns the default values for negotiable settings for sessions and connections, which represent the starting point for the storage array for negotiations.

Syntax

```
set storageArray icmpPingResponse=(TRUE | FALSE)
```

Set Storage Array iSNS Server IPv4 Address

This command sets the configuration method and address for an IPv4 Internet Storage Name Service (iSNS).

Syntax

```
set storageArray isnsIPv4ConfigurationMethod=[static | dhcp]
isnsIPv4Address=ipAddress
```

Set Storage Array iSNS Server IPv6 Address

This command sets the IPv6 address for the iSNS server.

Syntax

```
set storageArray isnsIPv6Address=ipAddress
```

Set Storage Array iSNS Server Listening Port

This command sets the iSNS server listening port.

Syntax

```
set storageArray isnsListeningPort=listeningPortIPAddress
```

Set Storage Array iSNS Server Refresh

This command refreshes the network address information for the iSNS server. This command is valid for only IPv4.

Syntax

```
set storageArray isnsServerRefresh
```

Set Storage Array Learn Cycle

This command sets the learn cycle for the battery backup unit. The learn cycle enables the storage management software to predict the remaining battery life. Learn cycles run at set intervals and store the results for software analysis.

Syntax

```
set storageArray learnCycleDate
(daysToNextLearnCycle=numberOfDays |
day=dayOfTheWeek) time=HH:MM
```

Set Storage Array Redundancy Mode

This command sets the redundancy mode of the storage array to either simplex or duplex.

Syntax

```
set storageArray redundancyMode=(simplex | duplex)
```

Set Storage Array Remote Status Notification

This command sets or changes the proxy configuration settings for the remote status notification feature. The proxy configuration settings are saved in the `devmgr.datadir\monitor\EMRSstate\EMRSRuntimeConfig.xml` file on the storage management station.

Syntax

```
set remoteStatusNotification proxyConfig
(PACProxy=proxyLocationURL | [proxyHost=hostURL] |
[proxyPort=hostPort])
```

Set Storage Array Security Key

Use this command to set the security key that is used throughout the storage array to implement the Drive Security premium feature. When any security-capable drive in the storage array is assigned to a secured pool, that drive will be security-enabled using the security key. Before you can set the security key, you must use the `create storageArray securityKey` command to create the security key.

Syntax

```
set storageArray securityKey
```

Set Storage Array Time

This command sets the clocks on both controllers in a storage array by synchronizing the controller clocks with the clock of the host from which you run this command.

Syntax

```
set storageArray time
```

Set Storage Array Module Positions

This command defines the position of the modules in a storage array. You must include all of the modules in the storage array when you enter this command.

Syntax

```
set storageArray modulePositions=(controller | moduleID ... moduleIDn)
```

Set Storage Array Unnamed Discovery Session

This command enables the storage array to participate in unnamed discovery sessions.

Syntax

```
set storageArray unnamedDiscoverySession=(TRUE | FALSE)
```

Set Module Alarm

This command turns on, turns off, or mutes the audible alarm for a specific module or all of the modules in a storage array.

Syntax

```
set (allModules | module [moduleID]
alarm=(enable | disable | mute))
```

Set Module Identification

This command sets the module ID of a controller module, an array module, or a drive module in a storage array. This command is valid only for controller modules, array modules, or drive modules that have module IDs that you can set through the controller firmware. You cannot use this command for controller modules, array modules, or drive modules that have a module ID that you set with a switch.

Syntax

```
set module ["serialNumber"] id=moduleID
```

Set Module Service Action Allowed Indicator

This command turns on or turns off the Service Action Allowed indicator light on a power-fan CRU, an interconnect-battery CRU, or an environmental services monitor (ESM) CRU. If the storage array does not support the Service Action Allowed indicator light feature, this command returns an error. If the storage array supports the command but is unable to turn on or turn off the indicator light, this command returns an error.

To turn on or turn off the Service Action Allowed indicator light on the controller CRU, use the `set controller serviceAllowedIndicator` command.

Syntax

```
set module [moduleID]
(powerFan [(left | right | top | bottom)] |
interconnect |
esm [(left | right | top | bottom)]) |
battery [(left | right)] |
serviceAllowedIndicator=(on | off)
```

Set Pool

This command defines the properties for a pool.

Syntax

```
set pool [poolName]
addDrives=(moduleID1,slotID1 ... moduleIDn,slotIDn)
raidLevel=(0 | 1 | 3 | 5 | 6)
owner=(a | b)
```

Set Pool Forced State

This command moves a pool into a Forced state. Use this command if the `start pool import` command does not move the pool to an Imported state or if the import operation does not work because of hardware errors. In a Forced state, the pool can be imported, and you can then identify the hardware errors.

Syntax

```
set pool [poolName] forcedState
```

Set Volume

This command defines the properties for a volume. You can use most parameters to define properties for one or more volumes. You also can use some parameters to define properties for only one volume. The syntax definitions are separated to show which parameters apply to several volumes and which apply to only one volume. Also, the syntax for volume mapping is listed separately.

NOTE In configurations where pools consist of more than 32 volumes, the operation can result in host I/O errors or internal controller reboots due to the expiration of the timeout period before the operation completes. If you experience host I/O errors or internal controller reboots, quiesce the host I/O and try the operation again.

Syntax Applicable to One or More Volumes

```
set (allVolumes | volume ["volumeName"] |
volumes ["volumeName1" ... "volumeNameN"] | volume <wwID>)
cacheFlushModifier=cacheFlushModifierValue
cacheWithoutBatteryEnabled=(TRUE | FALSE)
mediaScanEnabled=(TRUE | FALSE)
mirrorCacheEnabled=(TRUE | FALSE)
modificationPriority=(highest | high | medium | low | lowest)
owner=(a | b)
preReadRedundancyCheck=(TRUE | FALSE)
readCacheEnabled=(TRUE | FALSE)
writeCacheEnabled=(TRUE | FALSE)
cacheReadPrefetch=(TRUE | FALSE)
protectionInformationDisabled=(TRUE | FALSE)
```

Syntax Applicable to Only One Volume

```
set (volume ["volumeName"] | volume <wwID>)
addCapacity=volumeCapacity
[addDrives=(moduleID1,slotID1 ... moduleIDn,slotIDn)]
redundancyCheckEnabled=(TRUE | FALSE)
segmentSize=segmentSizeValue
userLabel=volumeName
preReadRedundancyCheck=(TRUE | FALSE)
```

Syntax Applicable to Volume Mapping

```
set (volume ["volumeName"] | volume <wwID> | accessVolume)
logicalUnitNumber=LUN
(host="hostName" |
hostGroup=("hostGroupName" | defaultGroup)
```

Set Volume Copy

This command defines the properties for a volume copy pair.

Syntax

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

Show Cache Backup Device Diagnostic Status

This command returns the status of backup device diagnostic tests started by the `start cacheBackupDevice diagnostic` command. If the diagnostics have finished, all of the results of the diagnostic tests are shown. If the diagnostics have not finished, only the results of the diagnostic tests that finished are shown. The results of the test are shown on the terminal, or you can write the results to a file.

Syntax

```
show cacheBackupDevice controller [(a | b)] diagnosticStatus [file="fileName"]
```

Show Cache Memory Diagnostic Status

This command returns the status of cache memory diagnostics started by the `start controller diagnostic` command. If the diagnostics have finished, all of the results of the diagnostic tests are shown. If all of the diagnostics have not finished, only the results of the diagnostic tests that finished are shown.

Syntax

```
show cacheMemory controller [(a | b)] diagnosticStatus file="fileName"
```

Show Controller

For each controller in a storage array, this command returns the following information:

- The status (Online or Offline)
- The current firmware and NVSRAM configuration
- The pending firmware configuration and NVSRAM configuration (if any)
- The board ID
- The product ID
- The product revision
- The serial number
- The date of manufacture
- The cache size or the processor size
- The date and the time to which the controller is set
- The associated volumes (including the preferred owner)
- The Ethernet port
- The physical disk interface
- The host interface, which applies only to Fibre Channel host interfaces

Syntax

```
show (allControllers | controller [(a | b)]) [summary]
```

Show Controller Diagnostic Status

This command returns the status of controller diagnostics started by the `start controller diagnostic` command. If the diagnostics have finished, the entire results of the diagnostic tests are shown. If the diagnostic tests have not finished, only the results of the of the tests that are finished are shown. The results of the test are shown on the terminal, or you can write the results to a file.

Syntax

```
show controller [(a | b)] diagnosticStatus [file=filename]
```

Show Controller NVSRAM

Syntax

```
show (allControllers | controller [(a | b)])
NVSRAM [hostType=hostTypeIndexLabel | host="hostName"]
```

Show Current iSCSI Sessions

This command returns information about an iSCSI session for either an iSCSI initiator or an iSCSI target.

Syntax

```
show iscsiInitiator ["initiatorName"] iscsiSessions
show iscsiTarget ["targetName"] iscsiSessions
```

Show Drive

For each drive in the storage array, this command returns the following information:

- The total number of drives
- The type of drive (Fibre Channel, SATA, or SAS)
- Information about the basic drive:
  - The module location and the slot location
  - The status
  - The capacity
  - The data transfer rate
  - The product ID
  - The firmware level
- Information about the drive channel:
  - The module location and the slot location
  - The preferred channel
  - The redundant channel
- Hot spare coverage
- Details for each drive

Syntax

```
show (allDrives
[driveMediaType=(HDD | SSD | unknown | allMedia)] |
[driveType=(fibre | SATA | SAS)]) |
drive [moduleID,slotID] |
drives [moduleID1,slotID1 ... moduleIDn,slotIDn])
summary
```

Show Drive Channel Statistics

This command shows the cumulative data transfer for the drive channel and error information. If the controller has automatically degraded a drive channel, this command also shows interval statistics. When you use this command, you can show information about one specific drive channel, several drive channels, or all drive channels.

Syntax

```
show (driveChannel [(1 | 2 | 3 | 4 | 5 | 6 | 7 | 8)] |
driveChannels [1 2 3 4 5 6 7 8] |
allDriveChannels) stats
```

Show Drive Download Progress

This command returns the status of firmware downloads for the drives that are targeted by the `download drive firmware` command or the `download storageArray driveFirmware` command.

Syntax

```
show allDrives downloadProgress
```

Show Host Interface Card Diagnostic Status

This command returns the status of running, interrupted, or completed host interface card diagnostics started by the `start hostCard diagnostic` command. If the diagnostics have finished, the entire results of the diagnostic tests are shown. If the diagnostics have not finished, only the results of the tests that are finished are shown. The results of the test are shown on the terminal, or you can write the results to a file.

Syntax

```
show hostCard controller [(a | b)] diagnosticStatus [progressOnly]
[file=filename]
```

Show Host Ports

For all of the host ports that are connected to a storage array, this command returns this information:

- The host port identifier
- The host port name
- The host type

Syntax

```
show allHostPorts
```

Show Data Replicator Software Volume Candidates

This command returns information about the candidate volumes on a remote storage array that you can use as secondary volumes in a Data Replicator Software configuration.

Syntax

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

Show Data Replicator Software Volume Synchronization Progress

This command returns the progress of data synchronization between the primary volume and the secondary volume in a Data Replicator Software configuration. This command shows the progress as a percentage of data synchronization that has been completed.

Syntax

```
show remoteMirror (localVolume ["volumeName"] |
localVolumes ["volumeName1" ... "volumeNameN"])
synchronizationProgress
```

Show Pool

This command returns this information about a pool:

- The status (Online or Offline)
- The drive type (Fibre Channel, SATA, or SAS)
- Module loss protection (yes or no)
- The current owner (the controller in slot A or the controller in slot B)
- The associated volumes and free capacity
- The associated drives

Syntax

```
show pool [poolName]
```

Show Pool Export Dependencies

This command shows a list of dependencies for the drives in a pool that you want to move from one storage array to a second storage array.

Syntax

```
show pool [poolName] exportDependencies
```

Show Pool Import Dependencies

This command shows a list of dependencies for the drives in a pool that you want to move from one storage array to a second storage array.

Syntax

```
show pool [poolName] importDependencies
[cancelImport=(TRUE | FALSE)]
```

Show Storage Array

This command returns configuration information about the storage array. The parameters return lists of values for the components and features in the storage array. You can enter the command with a single parameter or more than one parameter. If you enter the command without any parameters, the entire storage array profile is shown (which is the same information as if you entered the `profile` parameter).

Syntax

```
show storageArray | autoSupportConfig | profile |
batteryAge | connections | defaultHostType | healthStatus |
hostTypeTable | hotSpareCoverage | features | time |
volumeDistribution | longRunningOperations | summary
```

Show Storage Array Auto Configure

This command shows the default auto-configuration that the storage array creates if you run the `autoConfigure storageArray` command. If you want to determine whether the storage array can support specific properties, enter the parameter for the properties when you run this command. You do not need to enter any parameters for this command to return configuration information.

Syntax

```
show storageArray autoConfiguration
[driveType=(fibre | SATA | SAS)
raidLevel=(0 | 1 | 3 | 5 | 6)
poolWidth=numberOfDrives
poolCount=numberOfPools
volumesPerGroupCount=numberOfVolumesPerGroup
hotSpareCount=numberOfHotspares
segmentSize=segmentSizeValue
cacheReadPrefetch=(TRUE | FALSE)
securityType=(none | capable | enabled)]
```

Show Storage Array Host Topology

This command returns the storage domain topology, the host type labels, and the host type index for the host storage array.

Syntax

```
show storageArray hostTopology
```

Show Storage Array LUN Mappings

This command returns information from the storage array profile about the logical unit number (LUN) mappings in the storage array. Default group LUN mappings are always shown. If you run this command without any parameters, this command returns all of the LUN mappings.

Syntax

```
show storageArray lunMappings [host ["hostName"] |
hostgroup ["hostGroupName"]]
```

Show Storage Array Negotiation Defaults

This statement returns information about connection-level settings that are subject to initiator-target negotiation.

Syntax

```
show storageArray  iscsiNegotiationDefaults
```

Show Storage Array Remote Status Notification

This command shows the proxy configuration settings for the remote status notification feature that were defined by the `set remoteStatusNotification proxyConfig` command. The remote status proxy configuration settings apply to all of the storage arrays managed by the storage management station. The storage arrays must be capable of supporting the storage array profile and the support bundle. The proxy configuration settings are saved in the `devmgr.datadir\monitor\EMRSstate\EMRSRuntimeConfig.xml` file on the storage management station.

Syntax

```
show remoteStatusNotification proxyConfig
```

Show Storage Array Unconfigured iSCSI Initiators

This command returns a list of initiators that have been detected by the storage array but are not yet configured into the storage array topology.

Syntax

```
show storageArray unconfiguredIscsiInitiators
```

Show Storage Array Unreadable Sectors

This command returns a table of the addresses of all of the sectors in the storage array that cannot be read. The table is organized with column headings for the following information:

1. Volume user label
2. Logical unit number (LUN)
3. Accessible by (host or host group)
4. Date/time
5. Volume-relative logical block address (hexadecimal format – 0xnnnnnnnn)
6. Drive location (module t, slot s)
7. Drive-relative logical block address (hexadecimal format – 0xnnnnnnnn)
8. Failure type

The data is sorted first by the volume user label and second by the logical block address (LBA). Each entry in the table corresponds to a single sector.

Syntax

```
show storageArray unreadableSectors
```

Show String

This command shows a string of text from a script file. This command is similar to the `echo` command in MS-DOS and UNIX.

Syntax

```
show "textString"
```

Show Volume

For the volumes in a storage array, this command returns the following information:

- The number of volumes
- The name
- The status
- The capacity
- The RAID level
- The pool where the volume is located
- Details:
  - The volume ID
  - The subsystem ID
  - The drive type (Fibre Channel, SATA, or SAS)
  - Module loss protection
  - The preferred owner
  - The current owner
  - The segment size
  - The modification priority
  - The read cache status (enabled or disabled)
  - The write cache status (enabled or disabled)
  - The write cache without batteries status (enabled or disabled)
  - The write cache with mirroring status (enabled or disabled)
  - The flush write cache after time
  - The cache read prefetch setting (TRUE or FALSE)
  - The enable background media scan status (enabled or disabled)
  - The media scan with redundancy check status (enabled or disabled)
- The snapshot repository volumes
- The mirror repository volumes
- The snapshot volumes
- The snapshot copies

Syntax

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

Show Volume Action Progress

NOTE With firmware version 7.77, the `show volume actionProgress` command is deprecated. Replace this command with `show storageArray longRunningOperations`.

For a long-running operation that is currently running on a volume, this command returns information about the volume action and amount of the long-running operation 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).

Syntax

```
show volume ["volumeName"] actionProgress
```

Show Volume Copy

This command returns this information about volume copy operations:

- The copy status
- The start time stamp
- The completion time stamp
- The copy priority
- The source volume World Wide Identifier (WWID) or the target volume WWID
- The target volume Read-Only attribute setting

You can retrieve information about a specific volume copy pair or all of the volume copy pairs in the storage array.

Syntax

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

Show Volume Copy Source Candidates

This command returns information about the candidate volumes that you can use as the source for a volume copy operation.

Syntax

```
show volumeCopy sourceCandidates
```

Show Volume Copy Target Candidates

This command returns information about the candidate volumes that you can use as the target for a volume copy operation.

Syntax

```
show volumeCopy source ["sourceName"] targetCandidates
```

Show Volume Performance Statistics

This command returns information about the performance of the volumes in a storage array.

Syntax

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

Show Volume Reservations

This command returns information about the volumes that have persistent reservations.

Syntax

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

Start Cache Backup Device Diagnostic

ATTENTION Before you run this diagnostic test, make sure that the cache backup device has a status of Optimal.

This command runs diagnostic tests to evaluate the functionality of the device that you use to backup the data in the cache if you lose power to the controller. The diagnostic tests are specific to the backup device that is in the controller. Before you run these tests, make these changes to the controller that has the backup device on which you want to run diagnostics:

- Place the controller into service mode (use the `set controller [(a | b)] availability=serviceMode` command).
- Attach the management client directly to the controller through the management Ethernet port.

NOTE In a dual-controller configuration, you must run these diagnostic tests through the controller that you want to evaluate. You cannot run these diagnostic tests through the partner controller.

Syntax

```
start cacheBackupDevice [(1 | n | all)]
controller [(a | b)]
diagnostic diagnosticType=(basic | extended)
[extendedTestID=(writePatterns | random)]
```

Start Cache Memory Diagnostic

This command runs extended diagnostic tests to evaluate the functionality of the cache memory in a controller. Before you run these tests, you must make these changes to the controller on which you want to run diagnostics:

- Place the controller into Service mode (use the `set controller [(a | b)] availability=serviceMode` command).
- Attach the management client directly to the controller through the management Ethernet port.

NOTE In a dual controller configuration, you must run these diagnostic tests through the controller that you want to evaluate. You cannot run these diagnostic tests through the partner controller.

Syntax

```
start cacheMemory controller [(a | b)] diagnostic
diagnosticType=(basic | extended)
[extendedTestID=(marchC | patterns | psuedoRndm| DMACopy)]
```

Start Configuration Database Diagnostic

This command starts a diagnostic test to validate the configuration database in the controller firmware.

Syntax

```
start storageArray configDbDiagnostic
```

Start Controller Diagnostic

This command runs diagnostic tests to evaluate the functionality of the controller card. Before you run these tests, you must make these changes to the controller on which you want to run diagnostics:

- Place the controller into Service Mode (use the `set controller [(a | b)] availability=serviceMode` command).
- Attach the management client directly to the controller through the management Ethernet port.

NOTE In a dual controller configuration, you must run these diagnostic tests through the controller that you want to evaluate. You cannot run these diagnostic tests through the partner controller.

Syntax

```
start controller [(a | b)] diagnostic diagnosticType=(basic | extended)
[extendedTestID=(SRAM | FIFO | dataCopy| RAID5Parity | RAID6Parity)]
```

Start Controller Trace

This command starts an operation that saves debug trace information to a compressed file. The debug trace information can be used by a Sun Customer Care Center representative to help analyze how well a storage array is running.

Syntax

```
start controller [(a | b | both)] trace
dataType=(current | flushed | currentFlushed | all)
[forceFlush=(TRUE | FALSE)]
```

Start Drive Channel Fault Isolation Diagnostics

This command runs the drive channel fault isolation diagnostics and stores the results.

Syntax

```
start driveChannel [(1 | 2 | 3 | 4 | 5 | 6 | 7 | 8)]
controller [(a | b)] faultDiagnostics
(testDevices=[all | controller=(a | b) |
esms=[moduleID1 (left | right), ... , moduleIDn (left | right)] |
drives=[moduleID1, slotID1, ... , moduleIDn, slotIDn]] |
dataPattern=(fixed | pseudoRandom) |
patternNumber=[(0xhexadecimal | number)] |
maxErrorCount=integer |
testIterations=integer |
timeout=timeInterval)
```

Start Drive Channel Locate

This command identifies the drive modules that are connected to a specific drive channel by turning on the indicator lights for the drive module that is connected to the drive channel. (Use the `stop driveChannel locate` command to turn off the indicator lights on the drive module.)

Syntax

```
start driveChannel [(1 | 2 | 3 | 4 | 5 | 6 | 7 | 8)] locate
```

Start Drive Initialize

This command starts drive initialization.

ATTENTION Possible damage to the storage array configuration – As soon as you enter this command, all user data is destroyed.

Syntax

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

Start Drive Locate

This command locates a drive by turning on an indicator light on the drive. (Run the `stop drive locate` command to turn off the indicator light on the drive.)

Syntax

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

Start Drive Reconstruction

This command starts reconstructing a drive.

Syntax

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

Start Host Interface Card Diagnostic

This command runs diagnostic tests to evaluate the functionality of the controller host interface card. The diagnostic tests that this command runs are specific to the host interface card that is in the controller. Before you run these tests, you must make these changes to the controller that has the host interface card on which you want to run diagnostics:

- Place the controller into service mode (use the `set controller [(a | b)] availability=serviceMode` command).
- Attach the management client directly to the controller through the management Ethernet port.

NOTE In a dual controller configuration, you must run these diagnostic tests through the controller that you want to evaluate. You cannot run these diagnostic tests through the partner controller.

Syntax

```
start hostCard [(1 | 2 | 3 | 4)] controller [(a | b)] diagnostic
diagnosticType=(basic | extended)
[extendedTestID=(EDC | DMA | RAM | internalLoopback)]
```

Start iSCSI DHCP Refresh

This command initiates a refresh of the DHCP parameters for the iSCSI interface. If the configuration method for the interface is not set to DHCP, the procedure returns an error.

Syntax

```
start controller [(a | b)] iscsiHostPort [(1 | 2 | 3 | 4)] dhcpRefresh
```

Start Data Replicator Software Synchronization

This command starts Data Replicator Software synchronization.

Syntax

```
start remoteMirror primary ["volumeName"] synchronize
```

Start Module Locate

This command locates a module by turning on the indicator light. (Use the `stop module locate` command to turn off the indicator light for the module.)

Syntax

```
start module [moduleID] locate
```

Start Pool Defragment

This command starts a defragment operation on the specified pool.

NOTE Defragmenting a pool starts a long-running operation that you cannot stop.

Syntax

```
start pool [poolName] defragment
```

Start Pool Export

This command moves a pool into an Exported state. Then you can remove the drives that comprise the pool and reinstall the drives in a different storage array.

NOTE Within the pool, you cannot move volumes that are associated with the premium features from one storage array to another storage array.

Syntax

```
start pool [poolName] export
```

Start Pool Import

This command moves a pool into a Complete state to make a newly introduced pool available to its new storage array. The pool must be in an Exported state or a Forced state before you run this command. Upon successfully running the command, the pool is operational.

NOTE Within the pool, you cannot move volumes that are associated with the premium features from one storage array to another storage array.

Syntax

```
start pool [poolName] import
```

Start Pool Locate

This command identifies the drives that are logically grouped together to form the specified pool by blinking the indicator lights on the drives. (Use the `stop pool locate` command to turn off the indicator lights on the drives.)

Syntax

```
start pool [poolName] locate
```

Start Secure Drive Erase

This command erases all of the data from one or more Encryption Services (ES) drives so that they can be reused as ES drives. Run this command only when the ES drives are no longer part of a secure pool, or when the security key is unknown.

Syntax

```
start secureErase (drive [moduleID,slotID] |
drives [moduleID1,slotID1 ... moduleIDn,slotIDn])
```

Start Storage Array iSNS Server Refresh

This command initiates a refresh of the network address information for the iSNS server. If the DHCP server is marginal or unresponsive, the refresh operation can take from two to three minutes to complete.

NOTE This command is for IPv4 only.

Syntax

```
start storageArray isnsServerRefresh
```

Start Storage Array Locate

This command locates a storage array by turning on the indicator lights for the storage array. (Use the `stop storageArray locate` command to turn off the indicator lights for the storage array.)

Syntax

```
start storageArray locate
```

Start Volume Initialization

This command starts the formatting of a volume in a storage array.

NOTE Formatting a volume starts a long-running operation that you cannot stop.

Syntax

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

Stop Cache Backup Device Diagnostic

This command stops the cache backup device diagnostic tests that were started by the `start cacheBackupDevice diagnostic` command.

Syntax

```
stop cacheBackupDevice controller [(a | b)] diagnostic
```

Stop Cache Memory Diagnostic

This command stops the cache memory diagnostic tests that were started by the `start cacheMemory diagnostic` command.

Syntax

```
stop cacheMemory controller [(a | b)] diagnostic
```

Stop Configuration Database Diagnostic

This command stops the diagnostic test to validate the configuration database in the controller firmware that was started by the `start storageArray configDbDiagnostic` command.

Syntax

```
stop storageArray configDbDiagnostic
```

Stop Controller Diagnostic

This command stops the controller diagnostic tests that were started by the `start controller diagnostic` command.

Syntax

```
stop controller [(a | b)] diagnostic
```

Stop Drive Channel Fault Isolation Diagnostics

This command stops the drive channel fault isolation diagnostics, which stops the `start drive channel fault isolation diagnostics` command before it completes.

Syntax

```
stop driveChannel faultDiagnostics
```

Stop Drive Channel Locate

This command turns off the indicator lights on the drive modules that were turned on by the `start driveChannel locate` command.

Syntax

```
stop driveChannel locate
```

Stop Drive Locate

This command turns off the indicator light on the drive that was turned on by the `start drive locate` command.

Syntax

```
stop drive locate
```

Stop Host Interface Card Diagnostic

This command stops the host interface card diagnostic tests that were started by the `start host card diagnostic` command.

Syntax

```
stop host card controller [(a | b)] diagnostic
```

Stop Module Locate

This command turns off the indicator light on the module that was turned on by the `start module locate` command.

Syntax

```
stop module locate
```

Stop Pool Locate

This command turns off the indicator lights on the drives that were turned on by the `start pool locate` command.

Syntax

```
stop pool locate
```

Stop Snapshot

This command stops a copy-on-write operation.

Syntax

```
stop snapshot (volume [volumeName] |
volumes [volumeName1 ... volumeNameN])
```

Stop Storage Array Drive Firmware Download

This command stops a firmware download to the drives in a storage array that was started with the `download storageArray driveFirmware` command. This command does not stop a firmware download that is already in progress to a drive. This command stops all firmware downloads to drives that are waiting for the download.

Syntax

```
stop storageArray driveFirmwareDownload
```

Stop Storage Array iSCSI Session

This command forces the termination of a storage array iSCSI session.

Syntax

```
stop storageArray iscsiSession [sessionNumber]
```

Stop Storage Array Locate

This command turns off the indicator lights on the storage array that were turned on by the `start storageArray locate` command.

Syntax

```
stop storageArray locate
```

Stop Volume Copy

This command stops a volume copy operation.

Syntax

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

Suspend Remote Mirror

This command suspends a Data Replicator Software operation.

Syntax

```
suspend remoteMirror (primary [primaryVolumeName]
primaries [primaryVolumeName1 ... primaryVolumeNameN])
writeConsistency=(TRUE | FALSE)
```

Validate Storage Array Security Key

This command validates the security key for a storage array that has Encryption Services (ES) drives to make sure that the security key is not corrupt.

Syntax

```
validate storageArray securityKey
file="fileName"
passPhrase="passPhraseString"
```

Appendix A: Deprecated Commands and Parameters

This appendix lists the commands, the command formats, and the parameters that are no longer supported by this level of software. The information is presented in two tables. Table 1 lists commands that are no longer supported in this level of software and the new commands that replaced them. Table 2 lists the previous commands that are no longer supported in this level of software. Table 3 on page 80 lists the parameters that are no longer supported in this level of software and the new parameters that replaced them.

Deprecated Commands

Table 1 Commands Deprecated in Firmware Release 10.77

Deprecated Command	New Command
accept storageArray pendingTopology (allHosts | host [*user-label*] | hosts [*user-label*])	Removed.
create hostPort	The requirement to set the host type has been removed. The hostType parameter is used with the create host command.

Table 2 Deprecated Commands

Deprecated Command	New Command
create hostPort	The requirement to set the host type has been removed. The hostType parameter is used with the create host command.
create mapping volume=*userLabel* logicalGroupNumber=*logicalGroupNumber* [host | hostGroup]= *hostName* | *hostGroupName*	Use the set volume command to define the volume-to-LUN mapping.
create volume (drive | drives) [*moduleID1*,*slotID1* ... *moduleIDn*,*slotIDn*]	create volume drives=(*moduleID1*,*slotID1* ... *moduleIDn*,*slotIDn*) The new syntax for specifying drives requires an equal sign (=) after the drives parameter.
create volume driveCount [*numberOfDrives*]	create volume driveCount=*numberOfDrives* The new syntax for specifying the number of drives requires an equal sign (=) after the driveCount parameter.
create volume pool [*numberOfDrives*]	create volume pool=*poolName* The new syntax for specifying the pool name requires an equal sign (=) after the pool parameter.
delete mapping volume=*userLabel* [host | hostGroup]=*hostName* | *hostGroupName*	Use the remove volume LUNMapping command to remove a volume-to-LUN mapping.

Deprecated Command	New Command
disableSnapshot volume	Use the stop snapshot command to stop a copy-on-write operation.
download drive [*moduleID,slotID*] file=*filename*content=(firmware | modePage)	Use the download storageArray driveFirmware command to download the firmware images to all of the drives in the storage array.
download storageArray (firmwareFile | NVSRAMFile)=*filename*	download drive [*moduleID,slotID*] firmware file="*filename*" download storageArray firmware [, NVSRAM] file="*filename*" [, "*NVSRAM-filename*"] [downgrade=(TRUE | FALSE)] [activateNow=(TRUE | FALSE)] The new version of the storage management software provides unique commands to perform these functions.
download storageArray file=*filename* content=firmware [downgrade=(TRUE | FALSE)]	Use the download storageArray firmware command to download the firmware.
download storageArray file=*filename* content=NVSRAM	Use the download storageArray NVSRAM command to download the NVSRAM values.
download storageArray file=*filename* content=featureKey	Use the enable storageArray feature command to enable a premium feature.
download (allModules | module [*moduleID*]) file=*filename* content=firmware	Use the download (environmental card) firmware command to download the module firmware.
download module [0]	download allModules firmware file="*filename*" When you download ESM firmware to all of the drive modules, in the previous command "all modules" was defined by entering [0]. The new command uses the allModules parameter.
recreate storageArray mirrorRepository	The functionality is no longer supported.
recreateSnapshot volume	Use the recreate snapshot command to start a fresh copy-on-write operation by using an existing snapshot volume.
remove copyEntry target [*targetName*] [source [*sourceName*]]	Use the remove volumeCopy command to remove volume copy entries.
remove volumeReservations (allVolumes | volume [*volumeName*]	Use the clear volume command to clear persistent volume reservations.
set controller [(a | b)] batteryInstallDate=(TRUE | FALSE)	Use the reset storageArray batteryInstallDate command to reset the battery date.
set controller [(a | b)] NVSRAMByte [nvsram-offset]= (*nvsramByteSetting* | *nvsramBitSetting*)	set controller [(a | b)] globalNVSRAMByte [*nvsramOffset*= (*nvsramByteSetting* | *nvsramBitSetting*) This new command provides additional parameters for setting the NVSRAM values.

Deprecated Command	New Command
set controller [(a | b)] serviceMode=(TRUE | FALSE)	Use the set controller availability=serviceMode command to place the storage array in Service mode.
set drive [*moduleID,slotID*] operationalState=(optimal | failed)	Use the set drive operationalState=failed command to place a drive in the storage array in Failed mode. To return a drive to the Optimal state, use the revive drive command.
set hostPort	The requirement to set the host type has been removed. The hostType parameter is used with the Create Host statement.
set performanceMonitor interval=*intervalValue* iterations=*iterationValue*	Use the set sessions command to define values for the performance monitor interval and iterations.
set storageArray batteryInstallDate=(TRUE | FALSE)	Use the reset storageArray batteryInstallDate command to reset the battery date.
set storageArray clearEventLog=(TRUE | FALSE)	Use the clear storageArray eventLog command to clear the Event Log for the storage array.
set storageArray resetConfiguration=(TRUE | FALSE)	Use the clear storageArray configuration command to clear the entire configuration from the controllers in a storage array.
set storageArray RLSBaseline=currentTime	Use the reset storageArray RLSBaseline command to reset the read link status (RLS) baseline for all of the devices.
set storageArray dayOfTime= (TRUE | FALSE)	Use the set storageArray time command to set the clocks on both of the controllers in a storage array to the clock of the host.
set volume [*volumeName*] mirrorEnabled=(TRUE | FALSE)	Use the set volume command with mirror cache enabled.
set volumeCopy target [*targetName*] [source [*sourceName*]] priority=(lower | low | medium | high | highest)	Use the set volumeCopy command to define the volume copy pair.
set volumeLabel ID [*hexValue*] userLabel=*volumeName*	Use the set volume command to define a user name for a volume.
show hostTopology	Use the show storageArray hostTopology command to show all of the mappings, the storage domain topology, the host type labels, and the host type index for the host storage array.
show storageArray pendingTopology	Removed.
show storageArray preferredVolumeOwners	show storageArray profile This command, with the profile parameter, returns information about the preferred volume owner.
show volumes volume [*userLabel*]	show storageArray profile This command, with the profile parameter, returns information about the volume.

Deprecated Command	New Command
start increaseVolCapacity volume=*volumeName* incrementalCapacity= *capacityValue* drives=(*moduleID1,slotID1* ... *moduleIDn,slotIDn*)	Use the set volume command to define values for increasing the capacity of a volume.
start volumeCopy source=”*sourceName*” target=”*targetName*” copyPriority=(lowest | low | medium | high | highest)	Use the create volumeCopy command to create a volume copy pair and to start the volume copy.
upload storageArray file=*filename* content=configuration	Use the save configuration command to save a storage array configuration.
upload storageArray file=*filename* content=(allEvents | criticalEvents)	Use the save storageArray (allEvents | criticalEvents) command to save events to a file.
upload storageArray file=*filename* content=performanceStats	Use the save storageArray performanceStats command to save the performance statistics to a file.
upload storageArray file=*filename* content=RLSCounts	Use the save storageArray RLSCounts command to save the RLS counters to a file.
upload storageArray file=*filename* content=stateDump	Use the save storageArray stateCapture command to save state dumps to a file.
show volume actionProgress	Use the show storageArray longRunningOperations command to return information about the amount of a volume related long-running operation that is completed.

For information on how to handle errors and on how to define a password, use the set session command. See the “Set Session” command.

Deprecated Parameters

Table 3 Deprecated Parameters

Old Syntax	New Syntax
availability	Removed from the set pool command
bootp	Removed
clearEventLog	clear storageArray eventLog
copyEntry	volumeCopy
database	Removed
disableSnapshot	stop snapshot
enforceSoftLimit	Removed
featureKey	feature
filesystem	Removed
gatewayIPAddress	IPv4GatewayIP

Old Syntax	New Syntax
hostType	Removed from the create hostPort command and the set hostPort command.
id[]	volume<>
increaseVolCapacity	set volume addCapacity
incrementalCapacity	addCapacity
ipAddress	IPv4Address or IPv6Address
mapping	lunMapping
modePage	Removed
multimedia	Removed
on error	set session errorAction
performanceMonitor interval	performanceMonitorInterval
performanceMonitor iterations	performanceMonitorIterations
priority	copyPriority
-r	The -r terminal made a distinction between inband storage management and out-of-band storage management. the -r terminal is no longer required.
readAheadMultiplier	cacheReadPrefetch
recreateSnapshot	recreate snapshot
resetConfiguration	reset storageArray configuration
stateDump	stateCapture
subnetMask	IPv4SubnetMask
timeOfDay	time
upload	save
use password	set session password
volumeLabel	Removed
volumeReservations	show volume reservations or reservations

Please
Recycle

51336-00A