Command Line Interface and Script Commands

Version 10.77

May 2011

51334-00, Rev. A

Revision History

Version and Date	Description of Changes
51334-00, Rev. A, May 2011	Initial release of the document.

LSI Corporate Headquarters
Milpitas, CA
800-372-2447

Email
globalsupport@lsi.com

Website
www.lsi.com

Document Number: 51334-00, 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 CLI gives you direct access to a script engine that is a utility in the SANtricity ES Storage Manager software (also referred to as the *storage management software*). The script engine runs commands that configure and manage the storage arrays. The script engine reads the commands, or runs a script file, from the command line and performs the operations instructed by the commands.

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.

Terminal	Definition
`-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.
`-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* | all*object-types* | [*qualifier*]
(*object-type* [*identifier*] {*object-type* [*identifier*]} | *object-types* [*identifier-list*]))

An object can be identified in four ways:

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

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

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

Table 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* A drive is defined as two or three interger literal values separated by a comma.
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)
drive-channel-identifier (four drive ports per module)	(1 | 2 | 3 | 4)

Recurring Syntax	Syntax Value
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]]) | (repositoryPool=*user-label* [freeCapacityArea=*integer-literal*[2]]) 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*[3]] [moduleLossProtect=(TRUE | FALSE)[1]] | [protectionInformation=(none | enabled)[4]] |
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*[5] | hotSpareCount=*integer-literal* | segmentSize=*segment-size-spec* | cacheReadPrefetch=(TRUE | FALSE) securityType=(none | capable | enabled)[6] | protectionInformation=(none | enabled)[4]
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]To determine if a free capacity area exists, run the `show pool` command.

[3]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.

[4]The `protectionInformation` parameter applies to the drives in a volume group. Using the `protectionInformation` parameter, you can specify that protected drives must be selected for a pool. If you want to set the `protectionInformation` parameter to `enabled`, all of the drives in the pool must be capable of T10 protection information. You cannot have a mix of drives that are capable of T10 protection information and drives that are not capable of T10 protection information in the pool.

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

[6]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 – For the CLI commands that identify drive locations you need to specify the ID of the drive module and the ID of the slot in which a drive resides. 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)
```

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

Set Host Channel

Set Host Group

Set Host Port

Set iSCSI Initiator

Set iSCSI Target Properties

Show Current iSCSI Sessions

Show Host Ports

iSCSI Commands

Create iSCSI Initiator

Delete iSCSI Initiator

Reset Storage Array iSCSI Baseline

Save Storage Array iSCSI Statistics

Set iSCSI Initiator

Set iSCSI Target Properties

Show Current iSCSI Sessions

Show Storage Array Negotiation Defaults

Show Storage Array Unconfigured iSCSI Initiators

Start iSCSI DHCP Refresh

Stop Storage Array iSCSI Session

Module Commands

Download Environmental Card Firmware

Download Power Supply Firmware

Download Module Configuration Settings

Save Module Log

Set Module Alarm

Set Module Identification

Set Module Service Action Allowed Indicator

Start Module Locate

Stop Module Locate

Pool Commands

Create Pool

Delete Pool

Enable Pool Security

Session Command

Snapshot Commands

Storage Array Commands

Uncategorized Commands

Volume Commands

Volume Copy Commands

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)]
```

Parameters

Parameter	Description
`repositoryRAIDLevel`	The RAID level for the mirror repository volume. Valid values are `1`, `3`, `5`, or `6`.
`repositoryDrives`	The drives for the mirror repository volume. Specify the module ID value and the slot ID value for each drive that you assign to the mirror repository volume. Module ID values are `0` to `99`. Slot ID values are `1` to `32`. Enclose the module ID values and the slot ID values in parentheses.
`repositoryPoolUserLabel`	The alphanumeric identifier (including - and _) that you want to give the new pool in which the mirror repository volume will be loacated. Enclose the pool identifier in square brackets ([ ]).
`repositoryPool`	The name of the mirror repository pool where the mirror repository volume is located. (To determine the names of the pools in your storage array, run the `show storageArray profile` command.)
`driveMediaType`	The type of drive media that you want to use for the mirror repository pool. Valid drive media are these: ■ HDD – Use this option when you have hard drives in the drive module. ■ SSD – Use this option when you have solid state drives in the drive module. ■ `unknown` – Use if you are not sure what types of drive media are in the drive module. ■ `allMedia` – Use this option when you want to use all types of drive media that are in the drive module. Use this parameter when you use the `repositoryDriveCount` parameter. You must use this parameter when you have more than one type of drive media in your storage array.
`driveType`	The type of drive that you want to use in the mirror volume. You cannot mix drive types. You must use this parameter when you have more than one type of drive in your storage array. Valid drive types are : ■ `fibre` ■ SATA ■ SAS If you do not specify a drive type, the command defaults to `fibre`. Use this parameter when you use the `repositoryDriveCount` parameter.

Parameter	Description
`freeCapacityArea`	The index number of the free space in an existing pool that you want to use to create the mirror repository volume. Free capacity is defined as the free capacity between existing volumes in a pool. For example, a pool might have the following areas: volume 1, free capacity, volume 2, free capacity, volume 3, free capacity. To use the free capacity following volume 2, you would specify: `freeCapacityArea=2` Run the `show pool` command to determine if a free capacity area exists.
`repositoryDriveCount`	The number of unassigned drives that you want to use for the mirror repository volume.
`moduleLossProtect`	The setting to enforce module loss protection when you create the mirror repository volume. To enforce module loss protection, set this parameter to TRUE. The default value is FALSE.
`protectionInformation`	The setting to specify that a pool, and the volumes within the pool, has T10 protection information protection to make sure that the data maintains its integrity. When you use this parameter, only protected drives can be used for the pool. These settings are valid: ■ `none` – The pool does not have T10 protection information protection. ■ `enabled` – The pool has T10 protection information protection. The pool supports protected information and is formatted with protection information enabled.

Notes

If the drives that you select for the `repositoryDrives` parameter are not compatible with other parameters (such as the `repositoryRAIDLevel` parameter), the script command returns an error, and Data Replicator Software is not activated. The error returns the amount of space that is needed for the mirror repository volume. You can then re-enter the command, and specify the appropriate amount of space.

If you enter a value for the repository storage space that is too small for the mirror repository volumes, the controller firmware returns an error message that provides the amount of space that is needed for the mirror repository volumes. The command does not try to activate Data Replicator Software. You can re-enter the command by using the value from the error message for the repository storage space value.

When you assign the drives, if you set the `moduleLossProtect` parameter to TRUE and have selected more than one drive from any one module, the storage array returns an error. If you set the `moduleLossProtect` parameter to FALSE, the storage array performs operations, but the pool that you create might not have module loss protection.

When the controller firmware assigns the drives, if you set the `moduleLossProtect` parameter to TRUE, the storage array returns an error if the controller firmware cannot provide drives that result in the new pool having 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.

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"
```

Parameters

Parameter	Description
`userLabel`	The name of the HCA host port. Enclose the host port name in double quotation marks (" ").

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"
```

Parameters

Parameter	Description
`iscsiInitiator`	The name of the iSCSI initiator. Enclose the name in double quotation marks (" ").

Activate Storage Array Firmware

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

Syntax

```
activate storageArray firmware
```

Parameters

None.

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)]
```

Parameters

Parameter	Description
`driveType`	The type of drives that you want to use for the storage array. You must use this parameter when you have more than one type of drive in your storage array. Valid drive types are : ■ `fibre` ■ SATA ■ SAS If you do not specify a drive type, the command defaults to `fibre`.
`raidLevel`	The RAID level of the pool that contains the drives in the storage array. Valid RAID levels are `0`, `1`, 3, 5, or 6.
`poolWidth`	The number of drives in a pool in the storage array.
`poolCount`	The number of pools in the storage array. Use integer values.
`volumesPerGroupCount`	The number of equal-capacity volumes per pool. Use integer values.
`hotSpareCount`	The number of hot spares that you want in the storage array. Use integer values.
`segmentSize`	The amount of data (in KB) that the controller writes on a single drive in a volume before writing data on the next drive. Valid values are 8, 16, 32, 64, 128, 256, or 512.

Parameter	Description
`cacheReadPrefetch`	The setting to turn on or turn off cache read prefetch. To turn off cache read prefetch, set this parameter to FALSE. To turn on cache read prefetch, set this parameter to TRUE.
`securityType`	The setting to specify the security level when creating the pools and all associated volumes. These settings are valid: ■ `none` – The pool and volumes are not secure. ■ `capable` – The pool and volumes are capable of having security set, but security has not been enabled. ■ `enabled` – The pool and volumes have security enabled.
`protectionInformation`	The setting to specify that a pool, and the volumes within the pool, has T10 protection information protection to make sure that the data maintains its integrity. When you use this parameter, only protected drives can be used for the pool. These settings are valid: ■ `none` – The pool does not have T10 protection information protection. ■ `enabled` – The pool has T10 protection information protection. The pool supports protected information and is formatted with protection information enabled.

Notes

Drives and Pool

A pool is a set of drives that are logically grouped together by the controllers in the storage array. The number of drives in a pool is a limitation of the RAID level and the controller firmware. When you create a pool, follow these guidelines:

- Beginning with firmware version 7.10, you can create an empty pool so that you can reserve the capacity for later use.
- You cannot mix drive types, such as SAS, SATA and Fibre Channel, within a single pool.
- The maximum number of drives in a pool depends on these conditions:
  - The type of controller
  - The RAID level
- RAID levels include: 0, 1, 10, 3, 5, and 6.
  - In a 6140 or a 6140 storage array, a pool with RAID level 0 and a pool with RAID level 10 can have a maximum of 112 drives.
  - In a 6540 storage array, a pool with RAID level 0 and a pool with RAID level 10 can have a maximum of 224 drives.
  - A pool with RAID level 3, RAID level 5, or RAID level 6 cannot have more than 30 drives.
  - A pool with RAID level 6 must have a minimum of five drives.
  - If a pool with RAID level 1 has four or more drives, the storage management software automatically converts the pool to a RAID level 10, which is RAID level 1 + RAID level 0.
- If a pool contains drives that have different capacities, the overall capacity of the pool is based on the smallest capacity drive.
- To enable module loss protection, you must create a pool that uses drives located in at least three drive modules.

Hot Spares

Hot spare drives can replace any failed drive in the storage array. The hot spare must be the same type of drive as the drive that failed (that is, a SAS hot spare cannot replace a Fibre Channel drive). A hot spare must have capacity greater than or equal to any drive that can fail. If a hot spare is smaller than a failed drive, you cannot use the hot spare to rebuild the data from the failed drive. Hot spares are available only for RAID Level 1, RAID Level 3, RAID Level 5, or RAID Level 6.

Segment Size

The size of a segment determines how many data blocks that the controller writes on a single drive in a volume before writing data on the next drive. Each data block stores 512 bytes of data. A data block is the smallest unit of storage. The size of a segment determines how many data blocks that it contains. For example, an 8-KB segment holds 16 data blocks. A 64-KB segment holds 128 data blocks.

When you enter a value for the segment size, the value is checked against the supported values that are provided by the controller at run time. If the value that you entered is not valid, the controller returns a list of valid values. Using a single drive for a single request leaves other drives available to simultaneously service other requests.

If the volume is in an environment where a single user is transferring large units of data (such as multimedia), performance is maximized when a single data transfer request is serviced with a single data stripe. (A data stripe is the segment size that is multiplied by the number of drives in the pool that are used for data transfers.) In this case, multiple drives are used for the same request, but each drive is accessed only once. For optimal performance in a multiuser database or file system storage environment, set your segment size to minimize the number of drives that are required to satisfy a data transfer request.

Cache Read Prefetch

Cache read prefetch lets the controller copy additional data blocks into cache while the controller reads and copies data blocks that are requested by the host from the drive into cache. This action increases the chance that a future request for data can be fulfilled from cache. Cache read prefetch is important for multimedia applications that use sequential data transfers. The configuration settings for the storage array that you use determine the number of additional data blocks that the controller reads into cache. Valid values for the `cacheReadPrefetch` parameter are TRUE or FALSE.

Security Type

The `securityType` parameter is valid for drives that are capable of Encryption Services (ES). With ES, the controller firmware can create a key and activate the drive security feature. The drive security feature encrypts data as the data is written to the drive and decrypts the data as the data is read from the drive. Without the key created by the controller, the data written to the drive is inaccessible.

Before you can set the `securityType` parameter to `capable` or `enabled`, you must create a storage array security key. Use the `create storageArray securityKey` command to create a storage array security key. These commands are related to the security key:

- `create storageArray securityKey`
- `set storageArray securityKey`
- `import storageArray securityKey`
- `export storageArray securityKey`
- `start secureErase (drive | drives)`
- `enable pool [poolName] security`

Autoconfigure Storage Array Hot Spares

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

Syntax

```
autoConfigure storageArray hotSpares
```

Parameters

None.

Notes

When you run the `autoconfigure storageArray hotSpares` command, the controller firmware determines the number of hot spares to create based on the total number and type of drives in the storage array. For Fibre Channel drives, SATA drives, and SAS drives, the controller firmware creates one hot spare for the storage array and one additional hot spare for every 60 drives in the storage array.

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
```

Parameters

Parameter	Description
`localVolume`	The name of any remote-mirror volume. The remote-mirror volume can be the primary volume or the secondary volume of a remote mirrored pair. Enclose the volume name in square brackets ([ ]). If the volume name has special characters, you also must enclose the volume name in double quotation marks (" ").
`timeout`	The time interval within which the software can return the remote-mirror volume status. The timeout value is in minutes.

Notes

This command waits until the status becomes Optimal or the timeout interval expires. Use this command when you run the Asynchronous Data Replicator Software utility.

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)]
```

Parameters

Parameter	Description
`volume`	The name of the specific volume for which you want to check parity. Enclose the volume name in square brackets ([ ]). If the volume name has special characters, you also must enclose the volume name in double quotation marks (" ").
`parityErrorFile`	The file path and the file name to which you want to save the parity error information. Enclose the file name in double quotation marks (" "). For example: `file="C:\Program Files\CLI\logs\parerr.txt"` This command does not automatically append a file extension to the saved file. You must specify a file extension when entering the file name.
`mediaErrorFile`	The file path and the file name to which you want to save the media error information. Enclose the file name in double quotation marks (" "). For example: `file="C:\Program Files\CLI\logs\mederr.txt"` This command does not automatically append a file extension to the saved file. You must specify a file extension when entering the file name.
`priority`	The priority that the parity check has relative to host I/O activity. Valid values are `highest`, `high`, `medium`, `low`, or `lowest`.
`startingLBA`	The starting logical block address. Use integer values.
`endingLBA`	The ending logical block address. Use integer values.
`verbose`	The setting to capture progress details, such as percent complete, and to show the information as the volume parity is being repaired. To capture progress details, set this parameter to TRUE. To prevent capturing progress details, set this parameter to FALSE.

Notes

The starting logical block address and the ending logical block address are useful for very large single-volume LUNs. Running a volume parity check on a very large single volume LUN can take a long time. By defining the beginning address and ending address of the data blocks, you can reduce the time that a volume parity check takes to complete.

Clear Drive Channel Statistics

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

Syntax

```
clear all DriveChannels stats
```

Parameters

None.

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]
```

Parameters

Parameter	Description
None	If you do not enter a parameter, this command removes all configuration information for the storage array, except for information related to security and identification.
`all`	The setting to remove the entire configuration of the storage array, including security information and identification information. Removing all configuration information returns the storage array to its initial state.
`pools`	The setting to remove the volume configuration and the pool configuration. The rest of the configuration stays intact.

Notes

When you run this command, the storage array becomes unresponsive, and all script processing is canceled. You must remove and re-add the storage array to resume communication with the host. To remove an unresponsive storage array, access the Enterprise Management Window, and select **Edit >> Remove**. To re-add the storage array, access the Enterprise Management Window, select **Edit >> Add Storage Array**, and enter the appropriate IP addresses.

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
```

Parameters

None.

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
```

Parameters

None.

Clear Volume Reservations

This command clears persistent volume reservations.

Syntax

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

Parameters

Parameter	Description
`allVolumes`	The setting to clear persistent volume reservations on all of the volumes in the storage array.
`volume` or `volumes`	The name of the specific volume for which you want to clear persistent volume reservations. You can enter more than one volume name. Enclose the volume name in square brackets ([ ]). If the volume name has special characters, you also must enclose the volume name in double quotation marks (" ").

Notes

You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Names can have a maximum of 30 characters.

Clear Volume Unreadable Sectors

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

Syntax

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

Parameters

Parameter	Description
`allVolumes`	The setting to clear unreadable sector information from all of the volumes in the storage array.
`volume` or `volumes`	The name of the specific volume for which you want to clear unreadable sector information. You can enter more than one volume name. Enclose the volume name in square brackets ([ ]). If the volume name has special characters, you also must enclose the volume name in double quotation marks (" ").

Notes

You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Names can have a maximum of 30 characters.

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)]
```

Parameters

Parameter	Description
`userLabel`	The name that you want to give the host that you are creating. Enclose the host name in double quotation marks (" ").
`hostGroup`	The name of the host group in which you want to create a new host. Enclose the host group name in double quotation marks (" "). (If a host group does not exist, you can create a new host group by using the `create hostGroup` command.) The `defaultGroup` option is the host group that contains the host to which the volume is mapped.
`hostType`	The index label or the index number that identifies the host type. Use the `show storageArray hostTypeTable` command to generate a list of available host type identifiers. If the host type has special characters, enclose the host type in double quotation marks (" ").

Notes

You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Names can have a maximum of 30 characters.

A host is a computer that is attached to the storage array and accesses the volumes on the storage array through the host ports. You can define specific mappings to an individual host. You also can assign the host to a host group that shares access to one or more volumes.

A host group is an optional topological element that you can define if you want to designate a collection of hosts that share access to the same volumes. The host group is a logical entity. Define a host group only if you have two or more hosts that share access to the same volumes.

If you do not specify a host group in which to place the host that you are creating, the newly defined host belongs to the default host group.

Create Host Group

This command creates a new host group.

Syntax

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

Parameter

Parameter	Description
`userLabel`	The name that you want to give the host group that you are creating. Enclose the host name in double quotation marks (" ").

Notes

A host group is an optional topological element that you can define if you want to designate a collection of hosts that share access to the same volumes. The host group is a logical entity. Define a host group only if you have two or more hosts that can share access to the same volumes.

You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Names can have a maximum of 30 characters.

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)
```

Parameters

Parameter	Description
`identifier`	The 8-byte World Wide Identifier (WWID) or the 16-byte group identifier (GID) of the HBA or HCA host port. Enclose the WWID or the GID in double quotation marks (" ").
`userLabel`	The name that you want to give to the new HBA or HCA host port. Enclose the host port label in double quotation marks (" ").
`host`	The name of the host for which you are defining an HBA or HCA host port. Enclose the host name in double quotation marks (" ").
`interfaceType`	The identifier of the type of interface for the host port. The choices for the types of host port interfaces are: ■ FC – Fibre Channel ■ SAS – Serial-Attached SCSI ■ IB – Infiniband An FC or a SAS selection requires an 8-byte WWID. An IB selection requires a 16-byte group identifier (gid). If you do not specify the type of interface, FC is used as the default interface for the host port.

Notes

An HBA host port or an HCA host port is a physical connection on a host bus adapter or on a host channel adapter that resides in a host computer. An HBA host port or an HCA host port provides host access to the volumes in a storage array. If the HBA or the HCA has only one physical connection (one host port), the terms host port and host bus adapter or host channel adapter are synonymous.

You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Names can have a maximum of 30 characters.

Create iSCSI Initiator

This command creates a new iSCSI initiator object.

Syntax

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

Parameters

Parameters	Description
`iscsiName`	The default identifier of the iSCSI initiator. Enclose the identier in double quotation marks (" ").
`userLabel`	The name that you want to use for the iSCSI initiator. Enclose the name in double quotation marks (" ").
`host`	The name of the host in which the iSCSI initiator is installed. Enclose the name in double quotation marks (" ").
`chapSecret`	The security key that you want to use to authenticate a peer connection. Enclose the security key in double quotation marks (" ").

Notes

Challenge Handshake Authentication Protocol (CHAP) is a protocol that authenticates the peer of a connection. CHAP is based upon the peers sharing a *secret*. A secret is a security key that is similar to a password.

Use the `chapSecret` parameter to set up the security keys for initiators that require a mutual authentication.

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)]
```

Parameters

Parameter	Description
`drives`	The drives that you want to assign to the pool that you want to create. Specify the module ID value and the slot ID value for each drive that you assign to the pool. Module ID values are `0` to `99`. Slot ID values are `1` to `32`. Enclose the module ID values and the slot ID values in parentheses.
`raidLevel`	The RAID level of the pool that contains the volume. Valid values are `0`, `1`, `3`, `5`, or `6`.
`userLabel`	The alphanumeric identifier (including - and _) that you want to give the new pool. Enclose the pool identifier in double quotation marks (" ").
`driveMediaType`	The type of drive media that you want to use for the pool You must use this parameter when you have more than one type of drive media in your storage array. Valid drive media are: ■ HDD – Use this option when you have hard drives in the drive module. ■ SSD – Use this option when you have solid state drives in the drive module. ■ `unknown` – Use if you are not sure what types of drive media are in the drive module. ■ `allMedia` – Use this option when you want to use all types of drive media that are in the drive module.
`driveType`	The type of drive that you want to use in the volume. You cannot mix drive types. You must use this parameter when you have more than one type of drive in your storage array. Valid drive types are : ■ `fibre` ■ SATA ■ SAS If you do not specify a drive type, the command defaults to `fibre`.

Parameter	Description
moduleLossProtect	The setting to enforce module loss protection when you create the pool. To enforce module loss protection, set this parameter to TRUE. The default value is FALSE.
securityType	The setting to specify the security level when creating the pools and all associated volumes. These settings are valid: ■ none – The pool and volumes are not secure. ■ capable – The pool and volumes are capable of having security set, but security has not been enabled. ■ enabled – The pool and volumes have security enabled.
protectionInformation	The setting to specify that a pool, and the volumes within the pool, has T10 protection information protection to make sure that the data maintains its integrity. When you use this parameter, only protected drives can be used for the pool. These settings are valid: ■ none – The pool does not have T10 protection information protection. ■ enabled – The pool has T10 protection information protection. The pool supports protected information and is formatted with protection information enabled.

Notes

If you do not specify a capacity by using the capacity parameter, all of the drive capacity that is available in the pool is used. If you do not specify capacity units, bytes is used as the default value.

Cache Read Prefetch

The cacheReadPrefetch command lets the controller copy additional data blocks into cache while the controller reads and copies data blocks that are requested by the host from the drives into cache. This action increases the chance that a future request for data can be fulfilled from cache. Cache read prefetch is important for multimedia applications that use sequential data transfers. The configuration settings for the storage array that you use determine the number of additional data blocks that the controller reads into cache. Valid values for the cacheReadPrefetch parameter are TRUE or FALSE.

You do not need to enter a value for the cacheReadPrefetch parameter or the segmentSize parameter. If you do not enter a value, the controller firmware uses the usageHint parameter with fileSystem as the default value. Entering a value for the usageHint parameter and a value for the cacheReadPrefetch parameter or a value for the segmentSize parameter does not cause an error. The value that you enter for the cacheReadPrefetch parameter or the segmentSize parameter takes priority over the value for the usageHint parameter.

Segment Size

The size of a segment determines how many data blocks that the controller writes on a single drive in a volume before writing data on the next drive. Each data block stores 512 bytes of data. A data block is the smallest unit of storage. The size of a segment determines how many data blocks that it contains. For example, an 8-KB segment holds 16 data blocks. A 64-KB segment holds 128 data blocks.

When you enter a value for the segment size, the value is checked against the supported values that are provided by the controller at run time. If the value that you entered is not valid, the controller returns a list of valid values. Using a single drive for a single request leaves other drives available to simultaneously service other requests.

If the volume is in an environment where a single user is transferring large units of data (such as multimedia), performance is maximized when a single data transfer request is serviced with a single data stripe. A data stripe is the segment size that is multiplied by the number of drives in the pool that are used for data transfers. In this case, multiple drives are used for the same request, but each drive is accessed only once.

For optimal performance in a multiuser database or file system storage environment, set your segment size to minimize the number of drives that are required to satisfy a data transfer request.

Security Type

The `securityType` parameter is valid for drives that are capable of Encryption Services (ES). With ES, the controller firmware can create a key and activate the drive security feature. The drive security feature encrypts data as the data is written to the drive and decrypts the data as the data is read from the drive. Without the key created by the controller, the data written to the drive is inaccessible.

Before you can set the `securityType` parameter to `capable` or `enabled`, you must create a storage array security key. Use the `create storageArray securityKey` command to create a storage array security key. These commands are related to the security key:

- `create storageArray securityKey`
- `enable pool [poolName] security`
- `export storageArray securityKey`
- `import storageArray securityKey`
- `set storageArray securityKey`
- `start secureErase (drive | drives)`

Module Loss Protection

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

Module loss protection is not valid when you create volumes on existing pools.

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)]
```

Parameters

Parameter	Description
driveCount	The number of unassigned drives that you want to use in the pool.
poolUserLabel	The alphanumeric identifier (including – and _) that you want to give the new pool. Enclose the new pool name in double quotation marks (" ").
raidLevel	The RAID level of the pool that contains the volume. Valid values are 0, 1, 3, 5, or 6.
userLabel	The name that you want to give to the new volume. Enclose the new volume name in double quotation marks (" ").
driveMediaType	The type of drive media that you want to use for the pool. Valid drive media are these: ■ HDD – Use this option when you have hard drives in the drive module. ■ SSD – Use this option when you have solid state drives in the drive module. ■ unknown – Use if you are not sure what types of drive media are in the drive module. ■ allMedia – Use this option when you want to use all types of drive media that are in the drive module.
driveType	The type of drive that you want to use in the volume. You cannot mix drive types. You must use this parameter when you have more than one type of drive in your storage array. Valid drive types are : ■ fibre ■ SATA ■ SAS If you do not specify a drive type, the command defaults to fibre.
capacity	The size of the volume that you are adding to the storage array. Size is defined in units of bytes, KB, MB, GB, or TB.
owner	The controller that owns the volume. Valid controller identifiers are a or b, where a is the controller in slot A, and b is the controller in slot B. If you do not specify an owner, the controller firmware determines the owner.
cacheReadPrefetch	The setting to turn on or turn off cache read prefetch. To turn off cache read prefetch, set this parameter to FALSE. To turn on cache read prefetch, set this parameter to TRUE.
segmentSize	The amount of data (in KB) that the controller writes on a single drive in a volume before writing data on the next drive. Valid values are 8, 16, 32, 64, 128, 256, or 512.
usageHint	The setting for both cacheReadPrefetch parameter and the segmentSize parameter to be default values. The default values are based on the typical I/O usage pattern of the application that is using the volume. Valid values are fileSystem, dataBase, or multiMedia.
moduleLossProtect	The setting to enforce module loss protection when you create the pool. To enforce module loss protection, set this parameter to TRUE. The default value is FALSE.

Parameter	Description
dssPreAllocate	The setting to make sure that reserve capacity is allocated for future segment size increases. The default value is TRUE.
securityType	The setting to specify the security level when creating the pools and all associated volumes. These settings are valid: ■ none – The pool and volumes are not secure. ■ capable – The pool and volumes are capable of having security set, but security has not been enabled. ■ enabled – The pool and volumes have security enabled.
protectionInformation	The setting to specify that a pool, and the volumes within the pool, has T10 protection information protection to make sure that the data maintains its integrity. When you use this parameter, only protected drives can be used for the pool. These settings are valid: ■ none – The pool does not have T10 protection information protection. ■ enabled – The pool has T10 protection information protection. The pool supports protected information and is formatted with protection information enabled.

Notes

You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Names can have a maximum of 30 characters.

The driveCount parameter lets you choose the number of drives that you want to use in the pool. You do not need to specify the drives by module ID and slot ID. The controllers choose the specific drives to use for the pool.

The owner parameter defines which controller owns the volume.

If you do not specify a capacity using the capacity parameter, all of the drive capacity that is available in the pool is used. If you do not specify capacity units, bytes is used as the default value.

Cache Read Prefetch

Cache read prefetch lets the controller copy additional data blocks into cache while the controller reads and copies data blocks that are requested by the host from the drives into cache. This action increases the chance that a future request for data can be fulfilled from cache. Cache read prefetch is important for multimedia applications that use sequential data transfers. The configuration settings for the storage array that you use determine the number of additional data blocks that the controller reads into cache. Valid values for the cacheReadPrefetch parameter are TRUE or FALSE.

Segment Size

The size of a segment determines how many data blocks that the controller writes on a single drive in a volume before writing data on the next drive. Each data block stores 512 bytes of data. A data block is the smallest unit of storage. The size of a segment determines how many data blocks that it contains. For example, an 8-KB segment holds 16 data blocks. A 64-KB segment holds 128 data blocks.

When you enter a value for the segment size, the value is checked against the supported values that are provided by the controller at run time. If the value that you entered is not valid, the controller returns a list of valid values. Using a single drive for a single request leaves other drives available to simultaneously service other requests.

If the volume is in an environment where a single user is transferring large units of data (such as multimedia), performance is maximized when a single data transfer request is serviced with a single data stripe. A data stripe is the segment size that is multiplied by the number of drives in the pool that are used for data transfers. In this case, multiple drives are used for the same request, but each drive is accessed only once.

For optimal performance in a multiuser database or file system storage environment, set your segment size to minimize the number of drives that are required to satisfy a data transfer request.

You do not need to enter a value for the `cacheReadPrefetch` parameter or the `segmentSize` parameter. If you do not enter a value, the controller firmware uses the `usageHint` parameter with `fileSystem` as the default value. Entering a value for the `usageHint` parameter and a value for the `cacheReadPrefetch` parameter or a value for the `segmentSize` parameter does not cause an error. The value that you enter for the `cacheReadPrefetch` parameter or the `segmentSize` parameter takes priority over the value for the `usageHint` parameter.

Module Loss Protection

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

Module loss protection is not valid when you create volumes on existing pools.

Security Type

The `securityType` parameter is valid for drives that are capable of Encryption Services (ES). With ES, the controller firmware can create a key and activate the drive security feature. The drive security feature encrypts data as the data is written to the drive and decrypts the data as the data is read from the drive. Without the key created by the controller, the data written to the drive is inaccessible.

Before you can set the `securityType` parameter to `capable` or `enabled`, you must create a storage array security key. Use the `create storageArray securityKey` command to create a storage array security key. These commands are related to the security key:

- `create storageArray securityKey`
- `set storageArray securityKey`
- `import storageArray securityKey`
- `export storageArray securityKey`
- `start secureErase (drive | drives)`
- `enable pool [poolName] security`

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)]
```

Parameters

Parameter	Description
`pool`	The alphanumeric identifier (including - and _) for a specific pool in your storage array. Enclose the pool name in double quotation marks (" ").
`userLabel`	The name that you want to give the new volume. Enclose the new volume name in double quotation marks (" ").
`freeCapacityArea`	The index number of the free space in an existing pool that you want to use to create the new volume. Free capacity is defined as the free capacity between existing volumes in a pool. For example, a pool might have the following areas: volume 1, free capacity, volume 2, free capacity, volume 3, free capacity. To use the free capacity following volume 2, you would enter this index number: `freeCapacityArea=2` Run the `show pool` command to determine whether the free capacity area exists.
`capacity`	The size of the volume that you are adding to the storage array. Size is defined in units of `bytes`, KB, `MB`, GB, or TB.
`owner`	The controller that owns the volume. Valid controller identifiers are a or b, where a is the controller in slot A, and b is the controller in slot B. If you do not specify an owner, the controller firmware determines the owner.
`cacheReadPrefetch`	The setting to turn on or turn off cache read prefetch. To turn on cache read prefetch, set this parameter to TRUE. To turn off cache read prefetch, set this parameter to FALSE.
`segmentSize`	The amount of data (in KB) that the controller writes on a single drive in a volume before writing data on the next drive. Valid values are `8`, `16`, `32`, `64`, `128`, `256`, or `512`.
`usageHint`	The settings for both the `cacheReadPrefetch` parameter and the `segmentSize` parameter to be default values. The default values are based on the typical I/O usage pattern of the application that is using the volume. Valid values are `fileSystem`, `dataBase`, or `multiMedia`.
`dssPreAllocate`	The setting to make sure that reserve capacity is allocated for future segment size increases. The default value is TRUE.
`securityType`	The setting to specify the security level when creating the pools and all associated volumes. These settings are valid: `none` – The pool and volumes are not secure. `capable` – The pool and volumes are capable of having security set, but security has not been enabled. `enabled` – The pool and volumes have security enabled.
`protectionInformation`	The setting to specify that a pool, and the volumes within the pool, has T10 protection information protection to make sure that the data maintains its integrity. When you use this parameter, only protected drives can be used for the pool. These settings are valid: ■ `none` – The pool does not have T10 protection information protection. ■ `enabled` – The pool has T10 protection information protection. The pool supports protected information and is formatted with protection information enabled.

Notes

You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Names can have a maximum of 30 characters.

The `owner` parameter defines which controller owns the volume. The preferred controller ownership of a volume is the controller that currently owns the pool.

If you do not specify a capacity using the `capacity` parameter, all of the available capacity in the free capacity area of the pool is used. If you do not specify capacity units, `bytes` is used as the default value.

Segment Size

The size of a segment determines how many data blocks that the controller writes on a single drive in a volume before writing data on the next drive. Each data block stores 512 bytes of data. A data block is the smallest unit of storage. The size of a segment determines how many data blocks that it contains. For example, an 8-KB segment holds 16 data blocks. A 64-KB segment holds 128 data blocks.

When you enter a value for the segment size, the value is checked against the supported values that are provided by the controller at run time. If the value that you entered is not valid, the controller returns a list of valid values. Using a single drive for a single request leaves other drives available to simultaneously service other requests.

If the volume is in an environment where a single user is transferring large units of data (such as multimedia), performance is maximized when a single data transfer request is serviced with a single data stripe. A data stripe is the segment size that is multiplied by the number of drives in the pool that are used for data transfers. In this case, multiple drives are used for the same request, but each drive is accessed only once.

For optimal performance in a multiuser database or file system storage environment, set your segment size to minimize the number of drives that are required to satisfy a data transfer request.

Cache Read Prefetch

Cache read prefetch lets the controller copy additional data blocks into cache while the controller reads and copies data blocks that are requested by the host from the drives into cache. This action increases the chance that a future request for data can be fulfilled from cache. Cache read prefetch is important for multimedia applications that use sequential data transfers. The configuration settings for the storage array that you use determine the number of additional data blocks that the controller reads into cache. Valid values for the `cacheReadPrefetch` parameter are TRUE or FALSE. You do not need to enter a value for the `cacheReadPrefetch` parameter or the `segmentSize` parameter. If you do not enter a value, the controller firmware uses the `usageHint` parameter with `fileSystem` as the default value.

Entering a value for the `usageHint` parameter and a value for the `cacheReadPrefetch` parameter or a value for the `segmentSize` parameter does not cause an error. The value that you enter for the `cacheReadPrefetch` parameter or the `segmentSize` parameter takes priority over the value for the `usageHint` parameter.

Security Type

The `securityType` parameter is valid for drives that are capable of Encryption Services (ES). With ES, the controller firmware can create a key and activate the drive security feature. The drive security feature encrypts data as the data is written to the drive and decrypts the data as the data is read from the drive. Without the key created by the controller, the data written to the drive is inaccessible.

Before you can set the `securityType` parameter to `capable` or `enabled`, you must create a storage array security key. Use the `create storageArray securityKey` command to create a storage array security key. These commands are related to the security key:

- `create storageArray securityKey`
- `set storageArray securityKey`
- `import storageArray securityKey`
- `export storageArray securityKey`
- `start secureErase (drive | drives)`

- `enable pool [poolName] security`
- `create hostPort identifier`

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)]
```

Parameters

Parameter	Description
`drives`	The drives that you want to assign to the volume that you want to create. Specify the module ID value and the slot ID value for each drive that you assign to the volume. Module ID values are `0` to 99. Slot ID values are 1 to 32. Enclose the module ID values and the slot ID values in parentheses.
`poolUserLabel`	The alphanumeric identifier (including – and _) that you want to give the new pool. Enclose the pool identifier in double quotation marks (" ").
`raidLevel`	The RAID level of the pool that contains the volume. Valid values are `0`, 1, 3, 5, or 6.
`userLabel`	The name that you want to give the new volume. Enclose the new volume name in double quotation marks (" ").
`capacity`	The size of the volume that you are adding to the storage array. Size is defined in units of `bytes`, KB, MB, GB, or TB.
`owner`	The controller that owns the volume. Valid controller identifiers are a or b, where a is the controller in slot A, and b is the controller in slot B. If you do not specify an owner, the controller firmware determines the owner.
`cacheReadPrefetch`	The setting to turn on or turn off cache read prefetch. To turn off cache read prefetch, set this parameter to FALSE. To turn on cache read prefetch, set this parameter to TRUE.

Parameter	Description
segmentSize	The amount of data (in KB) that the controller writes on a single drive in a volume before writing data on the next drive. Valid values are 8, 16, 32, 64, 128, 256, or 512.
usageHint	The settings for both the cachReadPrefetch parameter and the segmentSize parameter to be default values. The default values are based on the typical I/O usage pattern of the application that is using the volume. Valid values are fileSystem, dataBase, or multiMedia.
moduleLossProtect	The setting to enforce module loss protection when you create the repository. To enforce module loss protection, set this parameter to TRUE. The default value is FALSE.
dssPreAllocate	The setting to make sure that reserve capacity is allocated for future segment size increases. This default value is TRUE.
securityType	The setting to specify the security level when creating the pools and all associated volumes. These settings are valid: ■ none – The pool and volumes are not secure. ■ capable – The pool and volumes are capable of having security set, but security has not been enabled. ■ enabled – The pool and volumes have security enabled.
protectionInformation	The setting to specify that a pool, and the volumes within the pool, has T10 protection information protection to make sure that the data maintains its integrity. When you use this parameter, only protected drives can be used for the pool. These settings are valid: ■ none – The pool does not have T10 protection information protection. ■ enabled – The pool has T10 protection information protection. The pool supports protected information and is formatted with protection information enabled.

Notes

If you set the raidLevel parameter to RAID 1, the controller firmware takes the list of drives and pairs them by using this algorithm:

Data drive	=	X
Parity drive	=	N/2 + X

In this algorithm X is 1 to N/2, and N is the number of drives in the list. For example, if you have six drives, the mirrored pairs are as follows:

Data	Parity
1	N/2 + 1 = 4
2	N/2 + 2 = 5
3	N/2 + 3 = 6

You can use any combination of alphanumeric characters, underscore (_), hyphen (-), and pound (#) for the names. Names can have a maximum of 30 characters.

The owner parameter defines which controller owns the volume. The preferred controller ownership of a volume is the controller that currently owns the pool.

If you do not specify a capacity using the `capacity` parameter, all of the drive capacity that is available in the pool is used. If you do not specify capacity units, `bytes` is used as the default value.

Segment Size

The size of a segment determines how many data blocks that the controller writes on a single drive in a volume before writing data on the next drive. Each data block stores 512 bytes of data. A data block is the smallest unit of storage. The size of a segment determines how many data blocks that it contains. For example, an 8-KB segment holds 16 data blocks. A 64-KB segment holds 128 data blocks.

When you enter a value for the segment size, the value is checked against the supported values that are provided by the controller at run time. If the value that you entered is not valid, the controller returns a list of valid values. Using a single drive for a single request leaves other drives available to simultaneously service other requests.

If the volume is in an environment where a single user is transferring large units of data (such as multimedia), performance is maximized when a single data transfer request is serviced with a single data stripe. A data stripe is the segment size that is multiplied by the number of drives in the pool that are used for data transfers. In this case, multiple drives are used for the same request, but each drive is accessed only once.

For optimal performance in a multiuser database or file system storage environment, set your segment size to minimize the number of drives that are required to satisfy a data transfer request.

Cache Read Prefetch

Cache read prefetch lets the controller copy additional data blocks into cache while the controller reads and copies data blocks that are requested by the host from the drive into cache. This action increases the chance that a future request for data can be fulfilled from cache. Cache read prefetch is important for multimedia applications that use sequential data transfers. The configuration settings for the storage array that you use determine the number of additional data blocks that the controller reads into cache. Valid values for the `cacheReadPrefetch` parameter are TRUE or FALSE.

You do not need to enter a value for the `cacheReadPrefetch` parameter or the `segmentSize` parameter. If you do not enter a value, the controller firmware uses the `usageHint` parameter with `fileSystem` as the default value. Entering a value for the `usageHint` parameter and a value for the `cacheReadPrefetch` parameter or a value for the `segmentSize` parameter does not cause an error. The value that you enter for the `cacheReadPrefetch` parameter or the `segmentSize` parameter takes priority over the value for the `usageHint` parameter.

Module Loss Protection

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

Module loss protection is not valid when you create volumes on existing pools.

Security Type

The `securityType` parameter is valid for drives that are capable of Encryption Services (ES). With ES, the controller firmware can create a key and activate the drive security feature. The drive security feature encrypts data as the data is written to the drive and decrypts the data as the data is read from the drive. Without the key created by the controller, the data written to the drive is inaccessible.

Before you can set the `securityType` parameter to `capable` or `enabled`, you must create a storage array security key. Use the `create storageArray securityKey` command to create a storage array security key. These commands are related to the security key:

- `create storageArray securityKey`
- `enable pool [poolName] security`
- `export storageArray securityKey`
- `import storageArray securityKey`

- `set storageArray securityKey`
- `start secureErase (drive | drives)`

Create Remote Mirror

This command creates both the primary volume and the secondary volume for a remote mirrored 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)]
```

Parameters

Parameter	Description
`primary`	The name of an existing volume on the local storage array that you want to use for the primary volume. Enclose the primary volume name in double quotation marks (" ").
`secondary`	The name of an existing volume on the remote storage array that you want to use for the secondary volume. Enclose the secondary volume name in double quotation marks (" ").
`remoteStorageArrayName`	The name of the remote storage array. Enclose the remote storage array name in double quotation marks (" ").
`remoteStorageArrayWwn`	The World Wide Identifier (WWID) of the remote storage array. Enclose the WWID in double quotation marks (" ").
`remotePassword`	The password for the remote storage array. Use this parameter when the remote storage array is password protected. Enclose the password in double quotation marks (" ").
`syncPriority`	The priority that full synchronization has relative to host I/O activity. Valid values are `highest`, `high`, `medium`, `low`, or `lowest`.
`autoResync`	The settings for automatic resynchronization between the primary volumes and the secondary volumes of a remote-mirror pair. This parameter has these values: ■ `enabled`– Automatic resynchronization is turned on. You do not need to do anything further to resynchronize the primary volume and the secondary volume. ■ `disabled`– Automatic resynchronization is turned off. To resynchronize the primary volume and the secondary volume, you must run the `resume remoteMirror` command.
`writeOrder`	The write order for data transmission between the primary volume and the secondary volume. Valid values are `preserved` or `notPreserved`.
`writeMode`	How the primary volume writes to the secondary volume. Valid values are `synchronous` or `asynchronous`.

Notes

You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Names can have a maximum of 30 characters.

When you choose the primary volume and the secondary volume, the secondary volume must be of equal or greater size than the primary volume. The RAID level of the secondary volume does not have to be the same as the primary volume.

Product shipments using the 6540 or 6580/6780 controller define a maximum of 128 remote mirrors. The 6140 and 6140 controllers can define a maximum of 64 remote mirrors.

Passwords are stored on each storage array in a management domain. If a password was not previously set, you do not need a password. The password can be any combination of a alphanumeric characters with a maximum of 30 characters. (You can define a storage array password by using the `set storageArray` command.)

Synchronization priority defines the amount of system resources that are used to synchronize the data between the primary volume and the secondary volume of a mirror relationship. If you select the highest priority level, the data synchronization uses the most system resources to perform the full synchronization, which decreases performance for host data transfers.

The `writeOrder` parameter applies only to asynchronous mirrors and makes them become part of a consistency group. Setting the `writeOrder` parameter to `preserved` causes the remote mirrored pair to transmit data from the primary volume to the secondary volume in the same order as the host writes to the primary volume. In the event of a transmission link failure, the data is buffered until a full synchronization can occur. This action can require additional system overhead to maintain the buffered data, which slows operations. Setting the `writeOrder` parameter to `notPreserved` frees the system from having to maintain data in a buffer, but it requires forcing a full synchronization to make sure that the secondary volume has the same data as the primary volume.

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)]
```

Parameters

Parameter	Description
`baseVolume`	The name of the base volume from which you want to take a snapshot. Enclose the base volume name in double quotation marks (" ").
`repositoryRAIDLevel`	Use this parameter when you create a new pool. The RAID level for the snapshot repository pool. Valid values are 1, 3, 5, or 6.
`repositoryDrives`	Use this parameter when you create a new pool. The drives that you want to assign to the snapshot repository pool. Specify the module ID value and the slot ID value for each drive that you assign to the snapshot repository volume. Module ID values are 0 to 99. Slot ID values are 1 to 32. Enclose the module ID values and the slot ID values in parentheses.
`repositoryDriveCount`	Use this parameter when you create a new pool. The number of unassigned drives that you want to use for the snapshot repository pool.
`repositoryPoolUserLabel`	Use this parameter when you create a new pool. The name of a new pool to be used for the repository volume. Enclose the repository pool name in double quotation marks (" ").

Parameter	Description
`repositoryPool`	The name of an existing pool where you want to place the repository volume. Use this parameter if you do not want to put the repository volume in the same pool as the base volume. The default is to use the same pool for both the base volume and the repository volume. Enclose the name of the repository pool in double quotation marks (" ").
`userLabel`	The name that you want to give to the snapshots volume. If you do not want to provide a name, the CLI creates a name using the base volume user label that you provide.
`moduleLossProtect`	The setting to enforce module loss protection when you create the snapshot repository volume. To enforce module loss protection, set this parameter to TRUE. The default value is FALSE.
`driveMediaType`	The type of drive medium that you want to use for the mirror repository volume. Valid drive media are these: ■ HDD – Use this option when you have hard drives in the drive module. ■ SSD – Use this option when you have solid state drives in the drive module. ■ `unknown` – Use if you are not sure what types of drive media are in the drive module. ■ `allMedia` – Use this option when you want to use all types of drive media that are in the drive module. Use this parameter when you use the `repositoryDriveCount` parameter. You must use this parameter when you have more than one type of drive media in your storage array.
`driveType`	The type of drive that you want to use in the volume. You cannot mix drive types. You must use this parameter when you have more than one type of drive in your storage array. Valid drive types are : ■ `fibre` ■ SATA ■ SAS If you do not specify a drive type, the command defaults to `fibre`. Use this parameter when you use the `repositoryDriveCount` parameter.
`freeCapacityArea`	The index number of the free space in an existing pool that you want to use to create the snapshot repository volume. Free capacity is defined as the free capacity between existing volumes in a pool. For example, a pool might have these areas: volume 1, free capacity, volume 2, free capacity, volume 3, free capacity. To use the free capacity following volume 2, you would specify: `freeCapacityArea=2` Run the `show pool` command to determine if a free capacity area exists.
`warningThresholdPercent`	The percentage of repository capacity at which you receive a warning that the snapshot repository volume is nearing full. Use integer values. For example, a value of `70` means 70 percent. The default value is `50`.

Parameter	Description
`repositoryPercentOfBase`	The size of the snapshot repository volume as a percentage of the base volume. Use integer values. For example, a value of `40` means 40 percent. The default value is `20`.
`repositoryUserLabel`	The name that you want to give to the snapshot repository volume. Enclose the snapshot repository volume name in double quotation marks (" ").
`repositoryFullPolicy`	How you want snapshot processing to continue if the snapshot repository volume is full. You can choose to fail writes to the base volume (`failBaseWrites`) or fail the snapshot volume (`failSnapshot`). The default value is `failSnapshot`.
`enableSchedule`	Use this parameter to turn on or to turn off the ability to schedule a snapshot operation. To turn on snapshot scheduling, set this parameter to TRUE. To turn off snapshot scheduling, set this parameter toFALSE.
`schedule`	Use this parameter to schedule a snapshot operation. You can use one of these options for setting a schedule for a snapshot operation: ■ `immediate` ■ `startDate` ■ `scheduleDay` ■ `startTime` ■ `scheduleInterval` ■ `endDate` ■ `noEndDate` ■ `timesPerDay` See the "Notes" section for information explaining how to use these options.

Notes

The volume that you are taking a snapshot of must be a standard volume in the storage array. The maximum number of snapshot volumes that you can create is one-half of the total number of volumes that are supported by a controller.

You can use any combination of alphanumeric characters, underscore (_), hyphen (-), and pound (#) for the names. Names can have a maximum of 30 characters.

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

If you do not choose a name for either the snapshot volume or the snapshot repository volume, the storage management software creates a default name by using the base volume name. An example of the snapshot volume name that the controllers might create is, if the base volume name is aaa and does not have a snapshot volume, the default snapshot volume name is aaa-1. If the base volume already has *n*-1 number of snapshot volumes, the default name is aaa-*n*. An example of the snapshot repository volume name that the controller might create is, if the base volume name is aaa and does not have a snapshot repository volume, the default snapshot repository volume name is aaa-R1. If the base volume already has *n*-1 number of snapshot repository volumes, the default name is aaa-R*n*.

If you do not specify the unconfigured space or free space, the snapshot repository volume is placed in the same pool as the base volume. If the pool where the base volume resides does not have enough space, this command fails.

Module Loss Protection

When you assign the drives, if you set the `moduleLossProtect` parameter to TRUE and have selected more than one drive from any one module, the storage array returns an error. If you set the `moduleLossProtect` parameter to FALSE, the storage array performs operations, but the pool that you create might not have module loss protection.

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

Scheduling Snapshots

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

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

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

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

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

Create Storage Array Security Key

This command creates a new security key for a storage array that has Encryption Services (ES) 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)
```

Parameters

Parameter	Description
`keyIdentifier`	A character string that you can read that is a wrapper around a security key. Enclose the key identifier in double quotation marks (" ").
`passPhrase`	A character string that encrypts the security key so that you can store the security key in an external file. Enclose the pass phrase in double quotation marks (" "). For information about the correct form for creating a valid pass phrase, refer to the Notes in this command description.
`file`	The file path and the file name to which you want to save the security key. For example: `file="C:\Program Files\CLI\sup\seckey.slk"` **IMPORTANT** – You must add a file extension of `.slk` to the end of the file name. Enclose the file path and name in double quotation marks (" ").
`commitSecurityKey`	This parameter commits the security key identifier to the storage array for all ES drives as well as the controllers. After the security key identifier is committed, a key is required to read data or write data. The data can only be read or changed by using a key, and the drive can never be used in a non-secure mode without rendering the data useless or totally erasing the drive.

Notes

Use this command for local key management only.

To use this command successfully, you need to have enough ES drives to create at least one pool.

The controller firmware creates a lock that restricts access to the ES drives. ES drives have a state called Security Capable. When you create a security key, the state is set to Security Enabled, which restricts access to all ES drives that exist within the storage array.

You can have a storage array configuration with more than one set of encrypted pools. Each pool can have a unique security key. The character string generated by the `keyIdentifier` parameter is a string that you can read and that enables you to identify the security key that you need. You can create a `keyIdentifer` by using one of these methods:

- You can enter up to 189 alphanumeric characters for a key identifier. The key identifier cannot have these characters:
  - White spaces
  - Punctuation
  - Symbols
- If you do not enter the `keyIdentifer` parameter, the controller automatically generates the `keyIdentifer` parameter.

Additional characters are automatically generated and appended to the end of the string that you enter for the key identifier. If you do not enter any string for the `keyIdentifier` parameter, the key identifier consists of only the characters that are automatically generated.

Your pass phrase must meet these criteria:

- The pass phrase must be between eight and 32 characters long.
- The pass phrase must contain at least one uppercase letter.
- The pass phrase must contain at least one lowercase letter.
- The pass phrase must contain at least one number.
- The pass phrase must contain at least one non-alphanumeric character, for example, < > @ +.

NOTE If your pass phrase does not meet these criteria, you will receive an error message and will be asked to retry the command.

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)]
```

Parameters

Parameter	Description
`source`	The name of an existing volume that you want to use as the source volume. Enclose the source volume name in double quotation marks (" ").
`target`	The name of an existing volume that you want to use as the target volume. Enclose the target volume name in double quotation marks (" ").
`copyPriority`	The priority that volume copy has relative to host I/O activity. Valid values are `highest`, `high`, `medium`, `low`, or `lowest`.
`targetReadOnlyEnabled`	The setting so that you can write to the target volume or only read from the target volume. To write to the target volume, set this parameter to FALSE. To prevent writing to the target volume, set this parameter to TRUE.
`copyType`	Use this parameter to create a volume copy with a snapshot. Creating a volume copy with a snapshot enables you to continue to write to the source volume while creating the volume copy. To create a volume copy with a snapshot, set this parameter to `online`. To create a volume copy without a snapshot, set this parameter to `offline`. If you do not use this parameter, the volume copy is created without a snapshot.

Parameter	Description
`repositoryPercentOfBase`	This parameter determines the size of the repository volume for the snapshot when you are creating a volume copy with a snapshot. The size of the repository volume is expressed as a percentage of the source volume, which is also called the base volume. Valid values for this parameter are `20`, `40`, `60`, `120`, and `default`. The default value is 20. If you do not use this parameter, the firmware uses a value of 20 percent. You must use the `copyType` parameter with the `repositoryPercentOfBase` parameter.
`repositoryGroupPreference`	This parameter determines to whichpool the snapshot repository volume is written. You have these choices: ■ `sameAsSource` – The snapshot repository volume is written to the same pool as the source volume if space is available. ■ `otherThanSource` – The snapshot repository volume is written to a different pool. Firmware determines which pool based on available space on the pools. ■ `default` – The snapshot repository volume is written to any pool that has space. For best performance, use the `sameAsSource` option. You must use the `copyType` parameter with the `repositoryGroupPreference` parameter.

Notes

You can use any combination of alphanumeric characters, happens, and underscores for the names. Names can have a maximum of 30 characters.

Copy priority defines the amount of system resources that are used to copy the data between the source volume and the target volume of a volume copy pair. If you select the highest priority level, the volume copy uses the most system resources to perform volume copy, which decreases performance for host data transfers.

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
```

Parameters

None.

Delete Host

This command deletes one or more hosts.

Syntax

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

Parameters

Parameter	Description
`host`	The name of the host that you want to delete. Enclose the host name in square brackets ([ ]). If the host name has special characters, you also must enclose the host name in double quotation marks (" ").
`hosts`	The names of several hosts that you want to delete. Enter the names of the hosts using these rules: ■ Enclose all of the names in square brackets ([ ]). ■ Enclose each of the names in double quotation marks (" "). ■ Separate each of the names with a space.

Notes

A host is a computer that is attached to the storage array and accesses the volumes on the storage array through the host ports on the host.

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]
```

Parameter

Parameter	Description
`hostGroup`	The name of the host group that you want to delete. Enclose the host group name in square brackets ([ ]). If the host group name has special characters, you also must enclose the host group name in double quotation marks (" ").

Notes

A host group is an optional topological element that is a collection of hosts that share access to the same volumes. The host group is a logical entity.

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]
```

Parameter

Parameter	Description
`hostPort`	The name of the host port that you want to delete. Enclose the name of the host port in square brackets ([ ]).

Notes

A host port is a physical connection on a host adapter that resides within a host computer. A host port provides a host access to the volumes in a storage array.

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]`

Parameter

Parameter	Description
`pool`	The alphanumeric identifier (including - and _) of the pool that you want to delete. Enclose the pool identifier in square brackets ([ ]).

Delete iSCSI Initiator

This command deletes a specific iSCSI initiator object.

Syntax

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

Parameters

Parameter	Description
`iscsiInitiator`	The identifier of the iSCSI initiator that you want to delete. The identifier of the iSCSI initiator can be either an iSCSI ID or a unique name. Enclose the identifier in double quotation marks (" "). You must also enclose the iscsiID in either square brackets ([ ]) or angle brackets (< >).

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]
```

Parameters

Parameter	Description
`volume` or `volumes`	The name of the snapshot volume that you want to delete. You can enter more than one snapshot volume name. Enclose the snapshot volume name in square brackets ([ ]). If the snapshot volume name has special characters, you also must enclose the snapshot volume name in double quotation marks (" ").
`schedule`	This parameter deletes the schedule for a specific snapshot volume. Only the schedule is deleted, the snapshot volume remains.

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)
```

Parameters

Parameter	Description
`allVolumes`	This parameter deletes all of the volumes in a storage array.
`volume` or `volumes`	The name of the volume that you want to delete. You can enter more than one volume name. Enclose the volume name in square brackets ([ ]). If the volume name has special characters, you also must enclose the volume name in double quotation marks (" ").
`removePool`	Deleting the last volume in a pool does not delete the pool. You can have a standalone pool (minus any volumes). To remove the standalone pool, set this parameter to TRUE. To keep standalone pools intact, set this parameter to FALSE.

Notes

When you use the `allVolumes` parameter, this command deletes volumes until all of the volumes are removed or until an error is encountered. If an error is encountered, this command does not try to delete the remaining volumes. Deleting volumes from different pools is possible. All of the pools that become empty are deleted if you set the `removePool` parameter to TRUE.

If you want to delete an entire pool, you can also use the `delete pool` command.

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"]
```

Parameters

Parameter	Description
`controller`	The controller on which you want to run the diagnostic tests. Valid controller identifiers are a or b, where a is the controller in slot A, and b is the controller in slot B. Enclose the controller identifier in square brackets ([ ]). If you do not specify a controller, the storage management software returns a syntax error.
`loopbackDriveChannel`	The drive channels on which you want to run the diagnostic tests. You can either choose to run the diagnostics on all channels or select a specific channel on which to run diagnostics. If you select a specific channel, valid values for the drive channels are 1, 2, 3, 4, 5, 6, 7, or 8.
`testID`	The identifier for the diagnostic test you want to run. The identifier and corresponding tests are as follows: ■ 1 – Read test ■ 2 – Write test ■ 3 – Data loop-back test ■ `discreteLines` – Discrete lines diagnostic test
`patternFile`	The file path and the file name that contains a data pattern that you want to use as test data. Enclose the file name of the data pattern in double quotation marks (" "). For example: `file="C:\Program Files\CLI\sup\patfile.txt"`

Notes

When you run a data loop-back test, you can optionally specify a file that contains a data pattern. If you do not specify a file, the controller firmware provides a default pattern.

Discrete lines are control lines and status lines that are connected between two controllers in a controller module. The discrete lines diagnostic test lets each controller check that control signal transitions can be observed at the control inputs of the alternate controller. The discrete lines diagnostic test automatically runs after each power-cycle or each controller-reset. You can run the discrete lines diagnostic test after you have replaced a component that failed the initial discrete lines diagnostic test. This test applies only to the 6540 controller module and the 6580/6780 controller module. The discrete lines diagnostic test returns one of these messages:

- When the discrete lines diagnostic test runs successfully, this message appears:

  ```
  The controller discrete lines successfully passed the diagnostic
  test. No failures were detected.
  ```

- If the discrete lines diagnostic test fails, this message appears:

  ```
  One or more controller discrete lines failed the diagnostic test.
  ```

- If the CLI cannot run the discrete lines diagnostic test, the CLI returns Error 270, which means that the discrete lines diagnostic test could not start nor complete.

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
```

Parameters

Parameter	Description
`controller`	The controller on which you want to run the cable diagnostic test. Valid controller identifiers are a or b, where a is the controller in slot A, and b is the controller in slot B. Enclose the controller identifier in square brackets ([ ]). If you do not specify a controller, the storage management software returns a syntax error.
`iscsiHostPorts`	The 8-byte World Wide Identifier (WWID) or the 16-byte group identifier (GID) of the HBA or HCA host port. Enclose the WWID or the GID in double quotation marks (" ").
`testID`	The identifier for the diagnostic test that you want to run. For this diagnostic test, the only choice is `cableDiagnostics`.

Notes

When you run the cable diagnostic test, the firmware returns the following information:

- Host Port: The port on which the diagnostic test was run.
- HIC: The host interface card associated with this port.
- The date and time the test was run.
- Status:
  - OK: All of the pairs of cables are good and do not have any faults.
  - Open: One or more of the four pairs of cables are open.
  - Short: One or more of the four pairs of cables are shorted.
  - Incomplete: One or more of the four pairs returned incomplete or invalid test results.
- Length – The Length of the cables are listed in meters and the following information about the cables is returned:
  - When the cable status is OK, the approximate lengths of the cable pairs are returned. The lengths of the cable paris are shown as a range (L1-L2), which are the shortest and the longest lengths of the cable pairs.
  - If the cable status is Open or Short, the approximate distance to the failure in the cable pairs. If there is one failure, the length is reported for that cable pair. If there is more than one failure, the information returned is both the shortest and longest lengths to the failures. The lengths are listed as a range (L1-L2) where L1<L2.

  - If the cable status is Incomplete, the information returned are the lengths for the shortest and longest cable pairs that the firmware can successfully test. The lengths are listed for the valid cable pairs as a range (L1-L2) where L1<L2.
- Register values for the cable diagnostic registers. The values are in a hexadecimal format:
  - Two bytes show the combined cable status (four bits per port).
  - Four two-byte numbers show the length of each channel.

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
```

Parameter

Parameter	Description
`primary` or `primaries`	The name of the primary volume of the remote mirror pair that you want to test. You can enter more than one primary volume name. Enclose the primary volume names in square brackets ([ ]). If the primary volume name has special characters, you also must enclose the primary volume name in double quotation marks (" ").

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"
```

Parameters

Parameter	Description
`file`	The file path and the file name that has the security key. For example: `file="C:\Program Files\CLI\sup\seckey.slk"` **IMPORTANT** – The file name must have an extension of `.slk`.
`passPhrase`	A character string that encrypts the security key so that you can store the security key in an external file.

Notes

Your pass phrase must meet these criteria:

- The pass phrase must be between eight and 32 characters long.

- The pass phrase must contain at least one uppercase letter.
- The pass phrase must contain at least one lowercase letter.
- The pass phrase must contain at least one number.
- The pass phrase must contain at least one non-alphanumeric character, for example, < > @ +.

NOTE If your pass phrase does not meet these criteria, you will receive an error message.

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)]
```

Parameters

None.

Notes

If you specify the `remoteMirror` parameter, this command disables the Data Replicator Software premium feature and takes away the structure of the mirror repository volume.

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`

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
```

Parameter

None.

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"
```

Parameters

Parameter	Description
`drive`	The drive to which you want to download the firmware image. Specify the module ID value and the slot ID value for each drive to which you want to download firmware. Module ID values are `0` to 99. Slot ID values are `1` to 32. Enclose the module ID values and the slot ID values in brackets ([ ]).
`file`	The file path and the file name of the file that contains the firmware image. Enclose the file path and the file name of the firmware image in double quotation marks (" "). For example: `file="C:\Program Files\CLI\dnld\drvfrm.dlp"` Valid file names have a `.dlp` extension.

Notes

Before trying to download drive firmware, take these precautions:

- Stop all I/O activity to the storage array before you download the firmware image. The `download drive` command blocks all I/O activity until the download finishes or fails; however, as a precaution, make sure that all I/O activity that might affect the drive is stopped.
- Make sure that the firmware image file is compatible with the drive module. If you download a firmware image file that is not compatible with the drive module that you have selected, the drive module might become unusable.
- Do not make any configuration changes to the storage array while you download drive firmware. Trying to make a configuration change can cause the firmware download to fail and make the selected drives unusable.

When you download the firmware to the drives, you must provide the full path and file name to the firmware image that is stored on your system.

You can use `download drive` command to test the firmware on one drive before you install the firmware on all of the drives in a storage array. The download returns one of these statuses:

- Successful
- Unsuccessful With Reason
- Never Attempted With Reason

Download Environmental Card Firmware

This command downloads environmental services monitor (ESM) firmware.

Syntax

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

Parameters

Parameter	Description
`allModule`	This parameter downloads new firmware to all of the modules in the storage array.
`module`	The drive module that contains the ESM card to which you want to load new firmware. Module ID values are `0` to 99. Enclose the module ID value in square brackets ([ ]).
`file`	The file path and the file name of the file that contains the firmware image. Enclose the file path and the file name of the firmware image in double quotation marks (" "). For example: `file="C:\Program Files\CLI\dnld\esmfrm.esm"` Valid file names have an `.esm` extension.

Notes

The `module` parameter downloads new firmware to a specific drive module. If you need to download new firmware to more than one drive module, but not all drive modules, you must enter this command for each drive module.

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
file="filename"
```

Parameters

Parameter	Description
`allModule`	This parameter downloads new firmware to all of the modules in the storage array.
`module`	The drive module that contains the ESM card to which you want to load new firmware. Module ID values are `0` to 99. Enclose the module ID value in square brackets ([ ]).
`file`	The file path and the file name of the file that contains the firmware image. Enclose the file path and the file name of the firmware image in double quotation marks (" "). For example: `file="C:\Program Files\CLI\dnld\moduleset.dlp"` Valid file names have a `.dlp` extension.

Notes

The `module` parameter downloads the factory default configuration settings to a specific drive module. If you need to download the factory default configuration settings to more than one drive module, but not all drive modules, you must enter this command for each drive module.

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]
```

Parameters

Parameter	Description
`allModules`	This parameter downloads new power supply firmware to all of the modules in the storage array.
`module` or `modules`	The module that contains the power supply to which you want to download new firmware. Module ID values are `0` to `99`. Enclose the module ID value in square brackets ([ ]).
`powerSupplyUpdate file`	The file path and the file name of the file that contains the firmware image. Enclose the file path and the file name of the firmware image in double quotation marks (" "). For example: `file="C:\Program Files\CLI\dnld\esmfrm.esm"` Valid file names have an `.esm` extension.
`powerSupplyUnit`	The power supply to which you want to download new firmware. Valid power supply identifiers are `left`, `right`, `top`, or `bottom`. Enclose the power-fan CRU identifier in square brackets ([ ])
`forceUpdate`	This parameter bypasses these checks: ■ To determine if the new firmware version is the same as the existing firmware version. ■ To determine if the power supply is in an Optimal state. ■ To determine if there is a redundant power supply that is in an Optimal state.

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"]
```

Parameter

Parameter	Description
`file`	The file path and the file name of the file that contains the firmware image. Enclose the file path and the file name of the firmware image in double quotation marks (" "). For example: `file="C:\Program Files\CLI\dnld\sadrvfrm.dlp"` Valid file names have a `.dlp` extension.

Notes

When you run this command, you can download more than one firmware image file to the drives in a storage array. The number of firmware image files that you can download depends on the storage array. The storage management software returns an error if you try to download more firmware image files than the storage array can accept.

You can schedule downloads for multiple drives at the same time, including multiple drives in a redundant pool. Each firmware image file contains information about the drive types on which the firmware image runs. The specified firmware images can be downloaded only to a compatible drive. Use the `download drive firmware` command to download a firmware image to a specific drive.

The `download storageArray driveFirmware` command blocks all I/O activity until either download try has been made for each candidate drive or you run the `stop storageArray downloadDriveFirmware` command. When the `download storageArray driveFirmware` command finishes downloading the firmware image, each candidate drive shows the download status for each drive. One of these statuses is returned:

- Successful
- Unsuccessful With Reason
- Never Attempted With Reason

Download Storage Array Firmware/NVSRAM

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

Syntax

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

Parameters

Parameter	Description
NVSRAM	The setting to download a file with NVSRAM values when you download a firmware file. Do not include square brackets with this parameter. Include a comma after the `firmware` parameter.
`file`	The file path and the file name that contains the firmware. Enclose the file path and the file name in double quotation marks (" "). For example: `file="C:\Program Files\CLI\dnld\safrm.dlp"` Valid file names have a `.dlp` extension.
`NVSRAM-filename`	The file path and the file name that contains the NVSRAM values. Enclose the NVSRAM file name in double quotation marks (" "). For example: `file="C:\Program Files\CLI\dnld\safrm.dlp"` Valid file names have a `.dlp` extension. Include a comma before the file name downloading both firmware and NVSRAM.
`downgrade`	The setting to load firmware that is a previous version. The default value is FALSE. Set the `downgrade` parameter to TRUE if you want to download an earlier version of firmware.
`activateNow`	The setting to activate the firmware image and the NVSRAM image. The default value is TRUE. If you set the `activateNow` parameter to FALSE, you must run the `activate storageArray firmware` command to activate the firmware values and the NVSRAM values at a later time.

Download Storage Array NVSRAM

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

Syntax

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

Parameter

Parameter	Description
`file`	The file path and the file name that contains the NVSRAM values. Enclose the NVSRAM file name in double quotation marks (" "). For example: `file="C:\Program Files\CLI\dnld\afrm.dlp"` Valid file names have a `.dlp` extension.

Enable Controller Data Transfer

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

Syntax

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

Parameter

Parameter	Description
`controller`	The controller that you want to revive. Valid controller identifiers are a or b, where a is the controller in slot A, and b is the controller in slot B. Enclose the controller identifier in square brackets ([ ]). If you do not specify a controller, the storage management software returns a syntax error.

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"
```

Parameters

Parameter	Description
`file`	The file path and the file name that has the security key. Enclose the file path and the file name that has the security key in double quotation marks (" "). For example: `file="C:\Program Files\CLI\sup\seckey.slk"` **IMPORTANT** – The file name must have an extension of `.slk`.
`passPhrase`	A character string that encrypts the security key so that you can store the security key in an external file. Enclose the pass phrase character string in double quotation marks (" ").

Notes

Your pass phrase must meet these criteria:

- The pass phrase must be between eight and 32 characters long.
- The pass phrase must contain at least one uppercase letter.
- The pass phrase must contain at least one lowercase letter.
- The pass phrase must contain at least one number.
- The pass phrase must contain at least one non-alphanumeric character, for example, < > @ +.

NOTE If your pass phrase does not meet these criteria, you will receive an error message.

Enable Pool Security

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

Syntax

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

Parameter

Parameter	Description
`pool`	The alphanumeric identifier (including - and _) of the pool that you want to place in the Security Enabled state. Enclose the pool identifier in square brackets ([ ]).

Notes

These conditions must be met to successfully run this command.

- All drives in the pool must be Encryption Services drives.
- The Drive Security premium feature must be enabled.
- The storage array security key has to be set.
- The pool is Optimal, and it does not have snapshot volumes or repository volumes.

The controller firmware creates a lock that restricts access to the ES drives. ES drives have a state called Security Capable. When you create a security key, the state is set to Security Enabled, which restricts access to all ES drives that exist within the storage array.

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"
```

Parameter

Parameter	Description
`file`	The file path and the file name of a valid feature key file. Enclose the file path and the file name in double quotation marks (" "). For example: `file="C:\Program Files\CLI\dnld\ftrkey.key"` Valid file names for feature key files end with a `.key` extension.

Notes

A premium feature is an additional application to enhance the capabilities of a storage array. The premium features are these:

- Storage domains
- Snapshots
- Data Replicator Software
- Mixed drive types
- High performance tier

- SSD support
- Drive Security

A feature pack is a predefined set of premium features, such as Storage Domains and Data Replicator Software. These premium features are combined for the convenience of the users.

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
```

Parameter

None.

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"
```

Parameters

Parameter	Description
`passPhrase`	A character string that encrypts the security key so that you can store the security key in an external file.
`file`	The file path and the file name to which you want to save the security key. For example: `file="C:\Program Files\CLI\sup\seckey.slk"` **IMPORTANT** – You must add a file extension of `.slk` to the end of the file name.

Notes

The storage array to which you will be moving drives must have drives with a capacity that is equal to or greater than the drives that you are importing.

The controller firmware creates a lock that restricts access to the Encryption Services (ES) drives. ES drives have a state called Security Capable. When you create a security key, the state is set to Security Enabled, which restricts access to all ES drives that exist within the storage array.

Your pass phrase must meet these criteria:

- The pass phrase must be between eight and 32 characters long.
- The pass phrase must contain at least one uppercase letter.

- The pass phrase must contain at least one lowercase letter.
- The pass phrase must contain at least one number.
- The pass phrase must contain at least one non-alphanumeric character, for example, < > @ +.

NOTE If your pass phrase does not meet these criteria, you will receive an error message and will be asked to retry the command.

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"
```

Parameters

Parameter	Description
`file`	The file path and the file name that has the original security key of the imported FDE drives. For example: `file="C:\Program Files\CLI\sup\seckey.slk"` **IMPORTANT** – The file that has the security key must have a file extension of `.slk`.
`passPhrase`	The character string that provides authentication for the security key. The pass phrase is 8 to 32 characters in length. You must use at least one number, one lowercase letter, one uppercase letter, and one non-alphanumeric character in the pass phrase. A space is not permitted.

Notes

The controller firmware creates a lock that restricts access to the ES drives. ES drives have a state called Security Capable. When you create a security key, the state is set to Security Enabled, which restricts access to all ES drives that exist within the storage array.

Your pass phrase must meet these criteria:

- The pass phrase must be between eight and 32 characters long.
- The pass phrase must contain at least one uppercase letter.
- The pass phrase must contain at least one lowercase letter.
- The pass phrase must contain at least one number.
- The pass phrase must contain at least one non-alphanumeric character, for example, < > @ +.

NOTE If your pass phrase does not meet these criteria, you will receive an error message and will be asked to retry the command.

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
```

Parameters

Parameter	Description
`file`	The file path and the file name of the DBM database you want to upload. Enclose the file name in double quotation marks (" "). For example: `file="C:\Array Backups\DBMbackup_03302010.dbm"` This command does not automatically append a file extension to the saved file. You must specify a file extension when entering the file name.
`validator`	The alphanumeric security code required to restore a storage array to an existing configuration. Use the `save storageArray dbmValidator` command to generate the required validation information XML file. After the validation information XML file is available, contact your Sun Customer Care Center representative to obtain the Validator.

Notes

It might take up to 30 minutes to restore controller functionality. Depending on the size of the database image, restoring the database might take up as much as 30 minutes. The host software will not show the controllers in an Optimal state until after all actions for loading the database image are completed on the controllers.

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)]
```

Parameters

Parameter	Description
`target`	The name of the target volume for which you want to reinitiate a volume copy operation. Enclose the target volume name in square brackets ([ ]). If the target volume name has special characters, you also must enclose the target volume name in double quotation marks (" ").
`source`	The name of the source volume for which you want to reinitiate a volume copy operation. Enclose the source volume name in square brackets ([ ]). If the source volume name has special characters, you also must enclose the source volume name in double quotation marks (" ").
`copyPriority`	The priority that the volume copy has relative to host I/O activity. Valid values are `highest`, `high`, `medium`, `low`, or `lowest`.
`targetReadOnlyEnabled`	The setting so that you can write to the target volume or only read from the target volume. To write to the target volume, set this parameter to FALSE. To prevent writing to the target volume, set this parameter to TRUE.
`copyType`	Use this parameter to create a volume copy with a snapshot. Creating a volume copy with a snapshot enables you to continue to write to the source volume while creating the volume copy. To reinitiate a volume copy with a snapshot, set this parameter to `online`. To reinitiate a volume copy with out a snapshot, set this parameter to `offline`.

Notes

Copy priority defines the amount of system resources that are used to copy the data between the source volume and the target volume of a volume copy pair. If you select the highest priority level, the volume copy uses the most system resources to perform the volume copy, which decreases performance for host data transfers.

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=subsystemVolumeID
[owner=(a | b)
cacheReadPrefetch=(TRUE | FALSE)
protectionInformation=(none | enabled)]
```

Parameters

Parameter	Description
`drive` or `drives`	The drives that you want to assign to the pool that will contain the volume that you want to recover. Specify the module ID value and the slot ID value for each drive that you assign to the volume. Module ID values are `0` to `99`. Slot ID values are `1` to `32`. Enclose the module ID values and the slot ID values insqurare brackets ([ ]).
`pool`	The name of an existing pool in which you want to create the volume. (To determine the names of the pools in your storage array, run the `show storageArray profile` command.)
`newPool`	The name that you want to give a new pool. Enclose the new pool name in double quotation marks (" ").
`userLabel`	The name of the volume that you want to recover. Enclose the volume name in double quotation marks (" ").
`volumeWWN`	The world wide name of the volume that you want to recover. The name is a 16 byte identifier, for example, 60080E500017B4320000000049887D77. Enclose the identifier in double quotation marks (" ").
`capacity`	The size of the volume that you are adding to the storage array. Size is defined in units of `bytes`, KB, MB, GB, or TB.
`offset`	The number of blocks from the start of the pool to the start of the referenced volume.
`raidLevel`	The RAID level of the pool that contains the drives. Valid values are `0`, `1`, 3, 5, or 6.
`segmentSize`	The amount of data (in KB) that the controller writes on a single drive in a pool before writing data on the next drive. Valid values are 8, 16, 32, 64, 128, 256, or 512.
`dssPreallocate`	The setting to turn on or turn off allocating volume storage capacity for future segment size changes. To turn on allocation, set this parameter to TRUE. To turn off allocation, set this parameter to FALSE.
`SSID`	The storage array subsystem identifier of a volume. Use the `show volume` command to determine the storage array subsystem identifier.

Parameter	Description
owner	The controller that owns the volume. Valid controller identifiers are a or b, where a is the controller in slot A, and b is the controller in slot B. If you do not specify an owner, the controller firmware determines the owner.
cacheReadPrefetch	The setting to turn on or turn off cache read prefetch. To turn off cache read prefetch, set this parameter to FALSE. To turn on cache read prefetch, set this parameter to TRUE.
protectionInformation	The setting to specify that a pool, and the volumes within the pool, has T10 protection information protection to make sure that the data maintains its integrity. When you use this parameter, only protected drives can be used for the pool. These settings are valid: ■ none – The pool does not have T10 protection information protection. ■ enabled – The pool has T10 protection information protection. The pool supports protected information and is formatted with protection information enabled.

Notes

The storage management software collects recovery profiles of the monitored storage arrays and saves the profiles on a storage management station.

If you attempt to recover a volume using the drive parameter or the drives parameter and the drives are in an unassigned state, the controller automatically creates a new pool. Use the newVolumeGroup parameter to specify a name for the new pool.

You can use any combination of alphanumeric characters,underscore (_), hyphen(-), and pound (#) for the names. Names can have a maximum of 30 characters.

The owner parameter defines which controller owns the volume. The preferred controller ownership of a volume is the controller that currently owns the pool.

Preallocating Storage Capacity

The dssPreallocate parameter enables you to assign capacity in a volume for storing information that is used to rebuild a volume. When you set the dssPreallocate parameter to TRUE, the storage space allocation logic in the controller firmware preallocates the space in a volume for future segment size changes. The preallocated space is the maximum allowable segment size. The dssPreallocate parameter is necessary for properly recovering volume configurations that are not retievable from the controller data base. To turn off the preallocation capability, setdssPreallocate to FALSE.

Segment Size

The size of a segment determines how many data blocks that the controller writes on a single drive in a volume before writing data on the next drive. Each data block stores 512 bytes of data. A data block is the smallest unit of storage. The size of a segment determines how many data blocks that it contains. For example, an 8-KB segment holds 16 data blocks. A 64-KB segment holds 128 data blocks.

When you enter a value for the segment size, the value is checked against the supported values that are provided by the controller at run time. If the value that you entered is not valid, the controller returns a list of valid values. Using a single drive for a single request leaves other drives available to simultaneously service other requests.

If the volume is in an environment where a single user is transferring large units of data (such as multimedia), performance is maximized when a single data transfer request is serviced with a single data stripe. (A data stripe is the segment size that is multiplied by the number of drives in the pool that are used for data transfers.) In this case, multiple drives are used for the same request, but each drive is accessed only once.

For optimal performance in a multiuser database or file system storage environment, set your segment size to minimize the number of drives that are required to satisfy a data transfer request.

Cache Read Prefetch

Cache read prefetch lets the controller copy additional data blocks into cache while the controller reads and copies data blocks that are requested by the host from disk into cache. This action increases the chance that a future request for data can be fulfilled from cache. Cache read prefetch is important for multimedia applications that use sequential data transfers. The configuration settings for the storage array that you use determine the number of additional data blocks that the controller reads into cache. Valid values for the `cacheReadPrefetch` parameter are TRUE or FALSE.

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"
```

Parameters

Parameter	Description
`passPhrase`	A character string that encrypts the security key so that you can store the security key in an external file.
`file`	The file path and the file name that has the security key. For example: `file="C:\Program Files\CLI\sup\seckey.slk"` **IMPORTANT** – The file name must have an extension of `.slk`.

Notes

Your pass phrase must meet these criteria:

- The pass phrase must be between eight and 32 characters long.
- The pass phrase must contain at least one uppercase letter.
- The pass phrase must contain at least one lowercase letter.
- The pass phrase must contain at least one number.
- The pass phrase must contain at least one non-alphanumeric character, for example, < > @ +.

NOTE If your pass phrase does not meet these criteria, you will receive an error message.

Re-create Data Replicator Software Repository Volume

NOTE With firmware version 7.80, the `recreate storageArray mirrorRepository` command is deprecated. This command is no longer supported in either the GUI or the CLI. If you attempts to run this command, an error message will be returned indicating that this functionality is no longer supported and that no changes will be made to the specified remote mirror repositories.

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)]
```

Parameters

Parameter	Description
`repositoryRAIDLevel`	The RAID level for the mirror repository volume. Valid values are `1`, 3, 5, or 6.
`repositoryDrives`	The drives for the mirror repository volume. Specify the module ID and slot ID for each drive that you assign to the mirror repository volume. Module ID values are `0` to `99`. Slot ID values are `1` to `32`. Enclose the module ID values and the slot ID values in parentheses.
`repositoryPool`	The name of the pool where the mirror repository volume is located.
`freeCapacityArea`	The index number of the free space in an existing pool that you want to use to re-create the mirror repository volume. Free capacity is defined as the free capacity between existing volumes in a pool. For example, a pool might have the following areas: volume 1, free capacity, volume 2, free capacity, volume 3, free capacity. To use the free capacity following volume 2, you would specify: `freeCapacityArea=2` Run the `show pool` command to determine if a free capacity area exists.
`repositoryDriveCount`	The number of unassigned drives that you want to use for the mirror repository volume.

Parameter	Description
`driveType`	The type of drive that you want to use for the mirror repository volume. You cannot mix drive types. You must use this parameter when you have more than one type of drive in your storage array. Valid drive types are : ■ `fibre` ■ SATA ■ SAS If you do not specify a drive type, the command defaults to `fibre`.
`moduleLossProtect`	The setting to enforce module loss protection when you create the mirror repository volume. To enforce module loss protection, set this parameter to TRUE. The default value is FALSE.
`protectionInformation`	The setting to specify that a pool, and the volumes within the pool, has T10 protection information protection to make sure that the data maintains its integrity. When you use this parameter, only protected drives can be used for the pool. These settings are valid: ■ `none` – The pool does not have T10 protection information protection. ■ `enabled` – The pool has T10 protection information protection. The pool supports protected information and is formatted with protection information enabled.

Notes

If you enter a value for the storage space of the mirror repository volume that is too small, the controller firmware returns an error message, which states the amount of space that is needed for the mirror repository volume. The command does not try to change the mirror repository volume. You can re-enter the command by using the value from the error message for the storage space value of the mirror repository volume.

When you assign the drives, if you set the `moduleLossProtect` parameter to TRUE and have selected more than one drive from any one module, the storage array returns an error. If you set the `moduleLossProtect` parameter to FALSE, the storage array performs operations, but the mirror repository volume that you create might not have module loss protection.

When the controller firmware assigns the drives, if you set the `moduleLossProtect` parameter to TRUE, the storage array returns an error if the controller firmware cannot provide drives that result in the new mirror repository volume having module loss protection. If you set the `moduleLossProtect` parameter to FALSE, the storage array performs the operation even if it means that the mirror repository volume might not have module loss protection.

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)]
```

Parameters

Parameter	Description
`volume` or `volumes`	The name of the specific volume for which you want to start a fresh copy-on-write operation. You can enter more than one volume name. Enclose the volume name in square brackets ([ ]). If the volume name has special characters, you must also enclose the volume name in double quotation marks (" ").
`userLabel`	The name of the snapshot volume. Enclose the snapshot volume name in double quotation marks (" "). If you enter more than one snapshot volume name, this command fails.
`warningThresholdPercent`	The percentage of repository capacity at which you receive a warning that the snapshot repository volume is nearing full. Use integer values. For example, a value of `70` means 70 percent. The default value is `50`.
`repositoryFullPolicy`	The type of processing that you want to continue if the snapshot repository volume is full. You can choose to fail writes to the base volume (`failBaseWrites`) or fail writes to the snapshot volume (`failSnapshot`). The default value is `failSnapshot`.

Notes

You can use any combination of alphanumeric characters, underscore (_), hyphen (-), and pound (#) for the names. Names can have a maximum of 30 characters.

If you do not specify a value for the `warningThresholdPercent` parameter or the `repositoryFullPolicy` parameter, the previously set value is used.

Recreating a Single Snapshot Volume or Multiple Snapshot Volumes with Optional Parameters

- If you specify one or more of the optional parameters, the re-create operation processes each snapshot volume separately.
- If you try to use the same user label for more than one volume, the command will fail.
- If you do not set the `warningThresholdPercent` parameter or the `repositoryFullPolicy` parameter, values that you previously set are used.

Recreating Multiple Snapshot Volumes without Optional Parameters

- If you list multiple snapshot volumes to be re-created but do not specify any of the optional parameters, the re-create operation processes the snapshot volumes as a "batch" process.
- Validation checks for the necessary snapshot-restarted preconditions are performed before restarting any snapshot. If any of the listed snapshot volumes fail the validation, the entire command fails and the snapshot volumes are not re-created. If the validation is successful for all of the snapshot volumes in the list, but one or more of the snapshots in the list fails to restart, the entire command fails and none of the snapshots are re-created.
- During snapshot re-creation, all affected volumes (snapshots, base, and repository) are appropriately quiesced and I/O operations are resumed to all affected volumes after all snapshots have been successfully re-created.

Remove Remote Mirror

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

Syntax

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

Parameter

Parameter	Description
`localVolume` or `localVolumes`	The name of the primary volume (the volume on the local storage array) that you want to remove. You can enter more than one volume name. Enclose the volume name in square brackets ([ ]). If the volume name has special characters, you also must enclose the volume name in double quotation marks (" ").

Remove Volume Copy

This command removes a volume copy pair.

Syntax

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

Parameters

Parameter	Description
`target`	The name of the target volume that you want to remove. Enclose the target volume name in square brackets ([ ]). If the target volume name has special characters, you also must enclose the target volume name in double quotation marks (" ").
`source`	The name of the source volume that you want to remove. Enclose the source volume name in square brackets ([ ]). If the source volume name has special characters, you also must enclose the source volume name in double quotation marks (" ").
`copyType`	Use this parameter to identify that a volume copy has a snapshot. If the volume copy has a snapshot, set this parameter to `online`. If the volume copy does not have a snapshot, set this parameter to `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))
```

Parameters

Parameter	Description
`allVolumes`	This parameter removes the LUN mapping from all of the volumes.
`volume` or `volumes`	The name of the specific volume that you want to remove from the LUN mapping. You can enter more than one volume name. Enclose the volume name in double quotation marks (" ") inside of square brackets ([ ]).
`accessVolume`	This parameter removes the access volume.
`host`	The name of the host to which the volume is mapped. Enclose the host name in double quotation marks (" ").
`hostGroup`	The name of the host group that contains the host to which the volume is mapped. Enclose the host group name in double quotation marks (" "). The `defaultGroup` value is the host group that contains the host to which the volume is mapped.

Notes

The access volume is the volume in a SAN environment that is used for communication between the storage management software and the storage array controller. The access volume uses a LUN address and consumes 20 MB of storage space that is not available for application data storage. An access volume is required only for in-band managed storage arrays.

ATTENTION Removing an access volume can damage your configuration – The agent uses the access volumes to communicate with a storage array. If you remove an access volume mapping for a storage array from a host that has an agent running on it, the storage management software is no longer able to manage the storage array through the agent.

You must use the `host` parameter and the `hostGroup` parameter when you specify a non-access volume or an access volume. The Script Engine ignores the `host` parameter or the `hostGroup` parameter when you use the `allVolumes` parameter or the `volumes` parameter.

Repair Volume Parity

This command repairs the parity errors on a volume.

Syntax

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

Parameters

Parameter	Description
`volume`	The name of the specific volume for which you want to repair parity. Enclose the volume name in square brackets ([ ]). If the volume name has special characters, you also must enclose the volume name in double quotation marks (" ").
`parityErrorFile`	The file path and the file name that contains the parity error information that you use to repair the errors. Enclose the file name in double quotation marks (" "). For example: `file="C:\Program Files\CLI\sup\parfile.txt"`
`verbose`	The setting to capture progress details, such as percent complete, and to show the information as the volume parity is being repaired. To capture progress details, set this parameter to TRUE. To prevent capturing progress details, set this parameter to 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
```

Parameters

Parameter	Description
`drive`	The location of the drive that you want to reconstruct. Specify the module ID value and the slot ID value of the drive that you want to revive. Module ID values are `0` to 99. Slot ID values are `1` to 32. Enclose the module ID value and the slot ID value in square brackets ([ ]).
`replacementDrive`	The location of the drive that you want to use for a replacement. Specify the module ID value and the slot ID value for the drive. Module ID values are `0` to 99. Slot ID values are `1` to 32.

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)]
```

Parameter

Parameter	Description
`controller`	The controller that you want to reset. Valid controller identifiers are a or b, where a is the controller in slot A, and b is the controller in slot B. Enclose the controller identifier in square brackets ([ ]). If you do not specify a controller owner, the controller firmware returns a syntax error.

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])
```

Parameters

Parameter	Description
`controller`	The controller that contains the battery for which you want to reset the age. Valid controller identifiers are a or b, where a is the controller in slot A, and b is the controller in slot B. Use the `controller` parameter only for controllers with batteries.
`batteryPack`	The battery pack contains both a left battery and a right battery. Valid identifiers are `left` or `right`, where `left` is the battery that supports the controller in slot A, and `right` is the battery that supports the controller in slot B. Use the `batteryPack` parameter only for controller modules with battery packs.

Notes

A controller might have a battery associated with it, so the controller is identified as either a or b. With the release of the 6580/6780 controller module, battery packs inside the interconnect-battery CRU are identified as either `left` or `right`. If the command statement uses the wrong parameter, an error appears.

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
```

Parameters

None.

Reset Storage Array Infiniband Statistics Baseline

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

Syntax

```
reset storageArray ibStatsBaseline
```

Parameters

None.

Notes

This command does not actually reset the raw counts maintained in the hardware and firmware. Instead, the firmware creates a snapshot of the current counter values and uses these values to report differences in the counts when the statistics are retrieved. The new baseline time is applied to both controllers so that the controller counts are synchronized with each other. If one controller resets without the other controller resetting, the counters are no longer synchronized. The client becomes aware that the controllers are not synchronized because the timestamp data reported along with the statistics is not the same for both controllers.

Reset Storage Array iSCSI Baseline

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

Syntax

```
reset storageArray iscsiStatsBaseline
```

Parameters

None.

Notes

This command resets the baseline to 0 for both controllers in the storage array. The purpose of resetting both of the controller baselines is to help make sure that the controller counts are synchronized between the controllers. If one controller resets but the second controller does not reset, the host is informed that the controllers are out of synchronization. The host is informed by the time stamps that are reported with the statistics.

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
```

Parameters

None.

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 `DrivePHY` errors. All other errors are deleted from the `.csv` file.

Syntax

```
reset storageArray SASPHYBaseline
```

Parameters

None.

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
```

Parameters

None.

Reset Storage Array Volume Distribution

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

Syntax

```
reset storageArray volumeDistribution
```

Parameters

None.

Notes

If you use this command on a host without a multi-path driver, you must stop I/O operations to the volumes until this command has completed to prevent application errors.

Under certain host operating system environments, you might be required to reconfigure the multi-path host driver. You might also need to make operating system modifications to recognize the new I/O path to the volumes.

Resume Remote Mirror

This command resumes a suspended Data Replicator Software operation.

Syntax

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

Parameters

Parameter	Description
`primary` or `primaries`	The name of the primary volume for which you want to resume operation. You can enter more than one primary volume name. Enclose the primary volume name in square brackets ([ ]). If the primary volume name has special characters, you also must enclose the primary volume name in double quotation marks (" ").
`writeConsistency`	The setting to identify the volumes in this command that are in a write-consistency group or are separate. For the volumes to be in the same write-consistency group, set this parameter to TRUE. For the volumes to be separate, set this parameter to FALSE.

Notes

If you set the `writeConsistency` parameter to TRUE, the volumes must be in a write-consistency group (or groups). This command resumes all write-consistency groups that contain the volumes. For example, if volumes A, B, and C are in a write-consistency group and they have remote counterparts A′, B′, and C′, the `resume remoteMirror volume ["A"] writeConsistency=`TRUE command resumes A-A′, B-B′, and C-C′.

Resume Snapshot Rollback

This command resumes a rollback operation that has entered a paused state. A rollback operation can enter a paused state due to processing errors, which will trigger a Needs Attention condition for the storage array.

If the rollback operation cannot be resumed, the selected snapshot volume reverts to a paused state, and the Needs Attention condition is displayed.

Syntax

```
resume rollback volume [snapshotVolumeName]
```

Parameter

Parameter	Description
`volume`	The name of the specific snapshot volume for which you want to resume a rollback operation. Enclose the snapshot volume name in square brackets ([ ]). If the snapshot volume name has special characters, you must also enclose the snapshot volume name in double quotation marks (" ").).

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]
```

Parameter

Parameter	Description
`drive`	The location of the drive that you want to revive. Specify the module ID value and the slot ID value of the drive that you want to revive. Module ID values are `0` to `99`. Slot ID values are `1` to `32`. Enclose the module ID value and the slot ID value in square brackets ([ ]).

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]
```

Parameter

Parameter	Description
`pool`	The alphanumeric identifier (including - and _) of the pool to be set to the Optimal state. Enclose the pool identifier in square brackets ([ ]).

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"
```

Parameters

Parameter	Description
`controller`	The controller with the NVSRAM values that you want to save. Valid controller identifiers are a or b, where a is the controller in slot A, and b is the controller in slot B. Enclose the controller identifier in square brackets ([ ]).
`file`	The file path and the file name to which you want to save the NVSRAM values. Enclose the NVSRAM file name in double quotation marks (" "). For example: `file="C:\Program Files\CLI\logs\nvsramb.txt"` This command does not automatically append a file extension to the saved file. You must specify a file extension when entering the file name.

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"
```

Parameter

Parameter	Description
`file`	The file path and the file name to which you want to save the results of the fault isolation diagnostics test on the drive channel. Enclose the file name in double quotation marks (" "). For example: `file="C:\Program Files\CLI\sup\fltdiag.bin"` This command does not automatically append a file extension to the saved file. You must specify a file extension when entering the file name.

Notes

A file extension is not automatically appended to the saved file. You must specify the applicable format file extension for the file. If you specify a file extension of `.txt`, the output will be in a text file format. If you specify a file extension of `.xml`, the output will be in an XML file format.

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"
```

Parameter

Parameter	Description
`logFile`	The file path and the file name to which you want to save the log sense data. Enclose the file name in double quotation marks (" "). For example: `file="C:\Program Files\CLI\logs\lgsendat.txt"` This command does not automatically append a file extension to the saved file. You must specify a file extension when entering the file name.

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"
```

Parameter

Parameter	Description
`logFile`	The file path and the file name to which you want to save the log sense data. Enclose the file name in double quotation marks (" "). For example: `file="C:\Program Files\CLI\logs\modulelogdat.txt"` This command does not automatically append a file extension to the saved file. You must specify a file extension when entering the file name.

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))]
```

Parameters

Parameter	Description
`file`	The file path and the file name to which you want to save the configuration settings. Enclose the file name in double quotation marks (" "). For example: `file="C:\Program Files\CLI\logs\saconf.cfg"` This command does not automatically append a file extension to the saved file. You must specify a file extension when entering the file name.
`allConfig`	The setting to save all of the configuration values to the file. (If you choose this parameter, all of the configuration parameters are set to TRUE.)
`globalSettings`	The setting to save the global settings to the file. To save the global settings, set this parameter to TRUE. To prevent saving the global settings, set this parameter to FALSE. The default value is TRUE.
`volumeConfigAndSettings`	The setting to save the volume configuration settings and all of the global settings to the file. To save the volume configuration settings and global settings, set this parameter to TRUE. To prevent saving the volume configuration settings and global settings, set this parameter to FALSE. The default value is TRUE.
`hostTopology`	The setting to save the host topology to the file. To save the host topology, set this parameter to TRUE. To prevent saving the host topology, set this parameter to FALSE. The default value is FALSE.
`lunMappings`	The setting to save the LUN mapping to the file. To save the LUN mapping, set this parameter to TRUE. To prevent saving the LUN mapping, set this parameter to FALSE. The default value is FALSE.

Notes

When you use this command, you can specify any combination of the parameters for the global setting, the volume configuration setting, the host topology, or the LUN mapping. If you want to enter all settings, use the `allConfig` parameter. The parameters are all optional.

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"
```

Parameter

Parameter	Description
`file`	The file path and the file name of the DBM database you want to save. Enclose the file name in double quotation marks (" "). For example: `file="C:\Array Backups \DBMbackup_03302010.dbm"` This command does not automatically append a file extension to the saved file. You must specify a file extension when entering the file name.

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"
```

Parameters

Parameter	Description
`file`	The file path and the file name of the DBM Validator required for Sun Customer Care Center. Enclose the file name in double quotation marks (" "). For example: `file="C:\Array Backups \DBMvalidator.xml"` This command does not automatically append a file extension to the saved file. You must specify a file extension when entering the file name.
`dbmDatabase`	The file path and the file name of the DBM database you want to use to restore a storage array. Enclose the file name in double quotation marks (" "). For example: `dbmDatabase="C:\Array Backups \DBMbackup_03302010.dbm"` This command does not automatically append a file extension to the saved file. You must specify a file extension when entering the file name.

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 | module)]
file="filename"
```

Parameters

Parameter	Description
`diagnosticData`	This parameter allows you to downloads the diagnostic data from either the controllers or the ESMs.
`file`	The file path and the file name to which you want to save the storage array diagnostic data. Enclose the file name in double quotation marks (" "). For example: `file="C:\Program Files\CLI\logs\sadiag.zip"` This command automatically saves the data to a compressed file; however, this command does not automatically append a file extension to the saved file. You must specify the `.zip` extension when entering the file name.

Notes

In versions of this command before 10.77, the user option was `esm` instead of `module`. Starting in 10.77 `module` replaces `esm`. The use of `esm` is still supported, but for best compatibility with future releases, replace `esm` with `module`.

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)]
```

Parameters

Parameter	Description
`allEvents`	The parameter to save all of the events to a file.
`criticalEvents`	The parameter to save only the critical events to a file.
`warningEvents`	The parameter to save only the warning events to a file.
`infoEvents`	The parameter to save only the informational events to a file.
`file`	The file path and the file name to which you want to save the events. Enclose the file name in double quotation marks (" "). For example: `file="C:\Program Files\CLI\logs\events.txt"` This command does not automatically append a file extension to the saved file. You must specify a file extension when entering the file name.
`count`	The number of events or critical events that you want to save to a file. If you do not enter a value for the count, all events or all critical events are saved to the file. If you enter a value for the count, only that number of events or critical events (starting with the last event entered) are saved to the file. Use integer values.
`forceSave`	The parameter to force saving the critical events to a file. To force saving the events, set this parameter to TRUE. The default value is FALSE.

Notes

You have the option to save all events (`allEvents`) or only the critical events (`criticalEvents`).

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"
```

Parameter

Parameter	Description
`file`	The file path and the file name to which you want to save the firmware inventory. Enclose the file name in double quotation marks (" "). For example: `file="C:\Program Files\CLI\logs\fwinvent.txt"` This command does not automatically append a file extension to the saved file. You must specify a file extension when entering the file name.

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"
```

Parameters

Parameter	Description
raw	The statistics that are collected are all statistics from the controller start-of-day. Enclose the parameter in square brackets ([ ]).
baseline	The statistics that are collected are all statistics from the time the controllers were reset to zero using the `reset storageArray ibStatsBaseline` command. Enclose the parameter in square brackets ([ ]).
file	The file path and the file name to which you want to save the performance statistics. Enclose the file name in double quotation marks (" "). For example: `file="C:\Program Files\CLI\sup\ibstat.txt"` This command does not automatically append a file extension to the saved file. You must specify a file extension when entering the file name.

Notes

If you have not reset the InfiniBand baseline statistics since the controller start-of-day, the time at the start-of-day is the default baseline time.

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"
```

Parameters

Parameter	Description
raw	The statistics collected are all statistics from the controller start-of-day. Enclose the parameter in square brackets ([ ]).
baseline	The statistics that are collected are all statistics from the time the controllers were reset to zero using the `reset storageArray ibStatsBaseline` command. Enclose the parameter in square brackets ([ ]).
file	The file path and the file name to which you want to save the performance statistics. Enclose the file name in double quotation marks (" "). For example: `file="C:\Program Files\CLI\logs\iscsistat.csv"` This command does not automatically append a file extension to the saved file. You can use any file name but you must use the `.csv` extension.

Notes

If you have not reset the iSCSI baseline statistics since the controller start-of-day, the time at the start-of-day is the default baseline time.

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"
```

Parameter

Parameter	Description
`file`	The file path and the file name to which you want to save the performance statistics. Enclose the file name in double quotation marks (" "). For example: `file="C:\Program Files\CLI\logs\sastat.csv"` This command does not automatically append a file extension to the saved file. You can use any file name, but you must use the `.csv` extension.

Save Storage Array RLS Counts

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

Syntax

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

Parameter

Parameter	Description
`file`	The file path and the file name to which you want to save the RLS counters. Enclose the file name in double quotation marks (" "). For example: `file="C:\Program Files\CLI\logs\rlscnt.csv"` The default name of the file that contains the RLS counts is `readLinkStatus.csv`. You can use any file name, but you must use the `.csv` extension.

Notes

To more effectively save RLS counters to a file, perform these steps:

1. Run the `reset storageArray RLSBaseline` command to set all of the RLS counters to 0.
2. Run the storage array for a predetermined amount of time (for instance, two hours).
3. Run the `save storageArray RLSCounts file="filename"` command.

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"
```

Parameter

Parameter	Description
`file`	The file path and the file name to which you want to save the SAS PHY counters. Enclose the file path and the file name in double quotation marks (" "). For example: `file="C:\Program Files\CLI\logs\sasphy.csv"` This command does not automatically append a file extension to the saved file. You can use any file name but you must use the `.csv` extension.

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"
```

Parameter

Parameter	Description
`file`	The file path and the file name to which you want to save the SOC error statistics. Enclose the file name in double quotation marks (" "). For example: `file="C:\Program Files\CLI\logs\socstat.csv"` The default name of the file that contains the SOC error statistics is `socStatistics.csv`. You can use any file name but you must use the `.csv` extension.

Notes

To more effectively save SOC error statistics to a file, perform these steps:

1. Run the `reset storageArray SOCBaseline` command to set all of the SOC counters to 0.
2. Run the storage array for a predetermined amount of time (for example, two hours).
3. Run the `save storageArray SOCCounts file="filename"` command.

Save Storage Array State Capture

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

Syntax

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

Parameter

Parameter	Description
`file`	The file path and the file name to which you want to save the state capture. Enclose the file name in double quotation marks (" "). For example: `file="C:\Program Files\CLI\logs\state.zip"` This command does not automatically append a file extension to the saved file. You must specify a file extension when entering the file name.

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"
```

Parameter

Parameter	Description
`file`	The file path and the file name to which you want to save the support-related data for the storage array. Enclose the file path and the file name in double quotation marks (" "). For example: `file="C:\Program Files\CLI\logs\supdat.zip"` This command does not automatically append a file extension to the saved file. You must specify a file extension when entering the file name.

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)
```

Parameters

Parameter	Description
`controller`	The controller for which you want to define properties. Valid identifiers for the controller are a or b, where a is the controller in slot A, and b is the controller in slot B. Enclose the identifier for the controller in square brackets ([ ]). If you do not specify a controller, the firmware for the controller returns a syntax error.
`availability`	The mode for the controller, which you can set to `online`, `offline`, or `serviceMode` (service).
`ethernetPort`	The attributes (options) for the management Ethernet ports. The entries to support this parameter are listed in the Syntax Element Statement Data table that follows. Many settings are possible, including setting the IP address, the gateway address, and the subnet mask address.
`globalNVSRAMByte`	A portion of the controller NVSRAM. Specify the region to be modified using the starting byte offset within the region and the byte value or bit value of the new data to be stored into the NVSRAM.
`hostNVSRAMByte`	The NVSRAM for the host-specific region. The setting specifies the host index for the specific host, the starting offset within the region, the number of bytes, and the the byte value or bit value of the new data to be stored into the NVSRAM.
`IPv4GatewayIP`	The IP address of the node that provides the interface to the network. The address format for the IPv4 gateway is **(0–255).(0–255).(0–255).(0–255)**
`IPv6RouterAddress`	The IP address of IPv6 router that connects two or more logical subnets. The address format for the IPv6 router is `(0–FFFF):(0–FFFF):(0–FFFF):(0–FFFF):(0–FFFF):(0–FFFF):(0–FFFF):(0–FFFF)`.

Parameter	Description
iscsiHostPort	The values that support this parameter are listed in the Syntax Element Statement Data table that follows. Many settings are possible, including setting the IP address, the gateway address, the subnet mask address, the IPv4 priority, and the IPv6 priority.
rloginEnabled	The setting for whether the remote login feature is turned on or turned off. To turn on the remote login feature, set this parameter to TRUE. To turn off the remote login feature, set this parameter to FALSE.
serviceAllowedIndicator	The setting for whether the Service Action Allowed indicator light is turned on or turned off. To turn on the Service Action Allowed indicator light, set this parameter to on. To turn off the Service Action Allowed indicator light, set this parameter to off.

Syntax Element Statement Data

Options for the ethernetPort Parameter
enableIPv4=(TRUE | FALSE) | enableIPv6=(TRUE | FALSE) | IPv6LocalAddress=(0-FFFF):(0-FFFF):(0-FFFF):(0-FFFF): (0-FFFF):(0-FFFF):(0-FFFF):(0-FFFF) | IPv6RoutableAddress=(0-FFFF):(0-FFFF):(0-FFFF):(0-FFFF): (0-FFFF):(0-FFFF):(0-FFFF):(0-FFFF) | IPv4Address=(0-255).(0-255).(0-255).(0-255) | IPv4ConfigurationMethod=[(static | dhcp)] | IPv4SubnetMask=(0-255).(0-255).(0-255).(0-255) | duplexMode=(TRUE | FALSE) | portSpeed=[(autoNegotiate | 10 | 100 | 1000)]

Options for the `iscsiHostPort` Parameter

```
IPv4Address=(0-255).(0-255).(0-255).(0-255) |
IPv6LocalAddress=(0-FFFF):(0-FFFF):(0-FFFF):(0-FFFF):
(0-FFFF):(0-FFFF):(0-FFFF):(0-FFFF) |
IPv6RoutableAddress=(0-FFFF):(0-FFFF):(0-FFFF):(0-FFFF):
(0-FFFF):(0-FFFF):(0-FFFF):(0-FFFF) |
IPv6RouterAddress=(0-FFFF):(0-FFFF):(0-FFFF):(0-FFFF):
(0-FFFF):(0-FFFF):(0-FFFF):(0-FFFF) |
enableIPv4=(TRUE | FALSE) |
enableIPv6=(TRUE | FALSE) |
enableIPv4Vlan=(TRUE | FALSE) |
enableIPv6Vlan=(TRUE | FALSE) |
enableIPv4Priority=(TRUE | FALSE) |
enableIPv6Priority=(TRUE | FALSE) |
IPv4ConfigurationMethod=(static | dhcp) |
IPv6ConfigurationMethod=(static | auto) |
IPv4GatewayIP=(TRUE | FALSE) |
IPv6HopLimit=[0-255] |
IPv6NdDetectDuplicateAddress=[0-256] |
IPv6NdReachableTime=[0-65535] |
IPv6NdRetransmitTime=[0-65535] |
IPv6NdTimeOut=[0-65535] |
IPv4Priority=[0-7] |
IPv6Priority=[0-7] |
IPv4SubnetMask=(0-255).(0-255).(0-255).(0-255) |
IPv4VlanId=[1-4094] |
IPv6VlanId=[1-4094] |
maxFramePayload=[frameSize] |
tcpListeningPort=[3260, 49152-65536] |
portSpeed=[(autoNegotiate | 1 | 10)]
```

Notes

NOTE Before firmware version 7.75, the `set controller` command supported an `NVSRAMByte` parameter. The `NVSRAMByte` parameter is deprecated and must be replaced with either the `hostNVSRAMByte` parameter or the `globalNVSRAMByte` parameter.

When you use this command, you can specify one or more of the parameters. You do not need to use all of the parameters.

Setting the `availability` parameter to `serviceMode` causes the alternate controller to take ownership of all of the volumes. The specified controller no longer has any volumes and refuses to take ownership of any more volumes. Service mode is persistent across reset cycles and power cycles until the `availability` parameter is set to `online`.

Use the `show controller NVSRAM` command to show the NVSRAM information. Before making any changes to the NVSRAM, contact your Sun Customer Care Center representative to learn what regions of the NVSRAM you can modify.

When the `duplexMode` option is set to TRUE, the selected Ethernet port is set to full duplex. The default value is half duplex (the `duplexMode` parameter is set to FALSE).

To make sure that the IPv4 settings or the IPv6 settings are applied, you must set these `iscsiHostPort` options:

- `enableIPV4=TRUE`
- `enableIPV6=TRUE`

The IPv6 address space is 128 bits. It is represented by eight 16-bit hexadecimal blocks separated by colons.

The `maxFramePayload` option is shared between IPv4 and IPv6. The payload portion of a standard Ethernet frame is set to `1500`, and a jumbo Ethernet frame is set to `9000`. When using jumbo frames, all of the devices that are in the network path should be capable of handling the larger frame size.

The `portSpeed` option is expressed as megabits per second (Mb/s).

Values for the `portSpeed` option of the `iscsiHostPort` parameter are in megabits per second (Mb/s).

The following values are the default values for the `iscsiHostOptions`:

- The `IPv6HopLimit` option is `64`.
- The `IPv6NdReachableTime` option is `30000` milliseconds.
- The `IPv6NdRetransmitTime` option is `1000` milliseconds.
- The `IPv6NdTimeOut` option is `30000` milliseconds.
- The `tcpListeningPort` option is `3260`.

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)
```

Parameters

Parameter	Description
`controller`	The controller that has the Service Action Allowed indicator light that you want to turn on or turn off. Valid controller identifiers are `a` or `b`, where `a` is the controller in slot A, and `b` is the controller in slot B. Enclose the controller identifier in square brackets ([ ]). If you do not specify a controller, the controller firmware returns a syntax error.
`serviceAllowedIndicator`	The setting to turn on or turn off the Service Action Allowed indicator light. To turn on the Service Action Allowed indicator light, set this parameter to `on`. To turn off the Service Action Allowed indicator light, set this parameter to `off`.

Notes

This command was originally defined for use with the 6540 controller module. This command is not supported by controller modules that were shipped before the introduction of the 6540 controller module. The 3992 and 3994 controllers also support this command.

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)
```

Parameters

Parameter	Description
`driveChannel`	The identifier number of the drive channel for which you want to set the status. Valid drive channel values are 1, 2, 3, 4, 5, 6, 7, or 8. Enclose the drive channel number in square brackets ([ ]).
`status`	The condition of the drive channel. You can set the drive channel status to `optimal` or `degraded`.

Notes

Use the `optimal` option to move a degraded drive channel back to the Optimal state. Use the `degraded` option when the drive channel is experiencing problems, and the storage array requires additional time for data transfers.

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)
```

Parameters

Parameter	Description
`drive` or `drives`	The location of the drive that you want to use for a hot spare. Specify the module ID value and the slot ID value for the drive. Module ID values are `0` to `99`. Slot ID values are `1` to `32`. Enclose the module ID value and the slot ID value in square brackets ([ ]).
`hotSpare`	The setting to assign the drive as the hot spare. To assign the drive as the hot spare, set this parameter to TRUE. To remove a hot spare assignment from a drive, set this parameter to 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)
```

Parameters

Parameter	Description
`drive` or `drives`	The location of the drive that you want to turn on or turn off the service action allowed indicator. Specify the module ID value and the slot ID value for the drive. Module ID values are `0` to `99`. Slot ID values are `1` to `32`. Enclose the module ID value and the slot ID value in square brackets ([ ]).
`serviceAllowedIndicator`	The setting to turn on or turn off the Service Action Allowed indicator light. To turn on the Service Action Allowed indicator light, set this parameter to `on`. To turn off the Service Action Allowed indicator light, set this parameter to `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
```

Parameter

Parameter	Description
`drive`	The location of the drive that you want to set to the Failed state. Specify the module ID value and the slot ID value for the drive. Module ID values are `0` to `99`. Slot ID values are `1` to `32`. Enclose the module ID value and the slot ID value in square brackets ([ ]).

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
```

Parameters

Parameter	Description
`drive` or `drives`	The location of the foreign drive that you want to add to the pool in a storage array. Specify the module ID value and the slot ID value for the drive. Module ID values are `0` to `99`. Slot ID values are `1` to `32`. Enclose the module ID value and the slot ID value in square brackets ([ ]).
`allDrives`	The setting to select all of the drives.

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)
```

Parameters

Parameter	Description
`host`	The name of the host that you want to assign to a host group. Enclose the host name in square brackets ([ ]). If the host name has special characters, you also must enclose the host name in double quotation marks (" ").
`hostGroup`	The name of the host group to which you want to assign the host. (The following table defines how the command runs if the host does or does not have individual mappings.) Enclose the host group name in double quotation marks (" "). The `defaultGroup` option is the host group that contains the host to which the volume is mapped.
`userLabel`	The new host name. Enclose the host name in double quotation marks (" ").
`hostType`	The index label or number of the host type for the host port. Use the `show storageArray hostTypeTable` command to generate a list of available host type identifiers. If the host type has special characters, enclose the host type in double quotation marks (" ").

Host Group Parameter	Host Has Individual Mappings	Host Does Not Have Individual Mappings
hostGroupName	The host is removed from the present host group and is placed under the new host group defined by *hostGroupName*.	The host is removed from the present host group and is placed under the new host group defined by *hostGroupName*.
none	The host is removed from the host group as an independent partition and is placed under the root node.	The host is removed from the present host group and is placed under the default group.
defaultGroup	The command fails.	The host is removed from the present host group and is placed under the default group.

Notes

When you use this command, you can specify one or more of the optional parameters.

For the names, you can use any combination of alphanumeric characters, hyphens, and underscores. Names can have a maximum of 30 characters.

Set Host Channel

This command defines the loop ID for the host channel.

Syntax

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

Parameters

Parameter	Description
hostChannel	The identifier number of the host channel for which you want to set the loop ID. Enclose the host channel identifier number in square brackets ([ ]). Use a host channel value that is appropriate for your particular controller model. A controller module might support one host channel or as many as eight host channels. Valid host channel values are a1, a2, a3, a4, a5, a6, a7, a8, b1, b2, b3, b4, b5, b6, b7, or b8.
preferredID	The port identifier for the specified host channel. Port ID values are 0 to 127.

Set Host Group

This command renames a host group.

Syntax

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

Parameters

Parameter	Description
`hostGroup`	The name of the host group that you want to rename. Enclose the host group name in square brackets ([ ]). If the host group name has special characters, you also must enclose the host group name in double quotation marks (" ").
`userLabel`	The new name for the host group. Enclose the new host group name in double quotation marks (" ").

Notes

You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Names can have a maximum of 30 characters.

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"
```

Parameters

Parameter	Description
`hostPort`	The name of the host port for which you want to change the host type, or for which you want to create a new name. Enclose the host port name in square brackets ([ ]). If the host port label has special characters, enclose the host port label in double quotation marks (" ").
`host`	The name of the host to which the host port is connected. Enclose the host name in double quotation marks (" ").
`userLabel`	The new name that you want to give to the host port. Enclose the new name of the host port in double quotation marks (" ").

Notes

When you use this command, you can specify one or more of the optional parameters.

You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Names can have a maximum of 30 characters.

Set iSCSI Initiator

This command sets the attributes for an iSCSI initiator.

Syntax

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

Parameters

Parameter	Description
`iscsiInitiator`	The name of the iSCSI initiator for which you want to set attributes. Enclose the iSCSI initiator name in double quotation marks (" ") and square brackets ([ ]).
`userLabel`	The new name that you want to use for the iSCSI initiator. Enclose the new name in double quotation marks (" ").
`host`	The name of the new host to which the host port is connected. Enclose the host name in double quotation marks (" ").
`chapSecret`	The security key that you want to use to authenticate a peer connection. Enclose the security key in double quotation marks (" ").

Notes

You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Names can have a maximum of 30 characters.

Challenge Handshake Authentication Protocol (CHAP) is a protocol that authenticates the peer of a connection. CHAP is based upon the peers sharing a *secret*. A secret is a security key that is similar to a password.

Use the `chapSecret` parameter to set up the security keys for initiators that require a mutual authentication. The CHAP secret must be between 12 characters and 57 characters. This table lists the valid characters.

Space	!	"	#	$	%	&	'	(	)	*	+
,	-	.	/	0	1	2	3	4	5	6	7
8	9	:	;	<	=	>	?	@	A	B	C
D	E	F	G	H	I	J	K	L	M	N	O
P	Q	R	S	T	U	V	W	X	Y	Z	[
\	]	^	_	'	a	b	c	d	e	f	g
h	i	j	k	l	m	n	o	p	q	r	s
t	u	v	w	x	y	z	{	|	}	~	

Set iSCSI Target Properties

This command defines properties for an iSCSI target.

Syntax

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

Parameters

Parameter	Description
`iscsiTarget`	The iSCSI target for which you want to set properties. Enclose the *`userLabel`* in double quotation marks (" "). You must also enclose the *`userLabel`* in either square brackets ([ ]) or angle brackets (< >).
`authenticationMethod`	The means of authenticating your iSCSI session.
`chapSecret`	The security key that you want to use to authenticate a peer connection.
`targetAlias`	The new name that you want to use for the target. Enclose the name in double quotation marks (" ").

Notes

Challenge Handshake Authentication Protocol (CHAP) is a protocol that authenticates the peer of a connection. CHAP is based upon the peers sharing a *secret*. A secret is a security key that is similar to a password.

Use the `chapSecret` parameter to set up the security keys for initiators that require a mutual authentication. The CHAP secret must be between 12 characters and 57 characters. This table lists the valid characters.

Space	!	"	#	$	%	&	'	(	)	*	+
,	-	.	/	0	1	2	3	4	5	6	7
8	9	:	;	<	=	>	?	@	A	B	C
D	E	F	G	H	I	J	K	L	M	N	O
P	Q	R	S	T	U	V	W	X	Y	Z	[
\	]	^	_	'	a	b	c	d	e	f	g
h	i	j	k	l	m	n	o	p	q	r	s
t	u	v	w	x	y	z	{	|	}	~	

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))
```

Parameters

Parameter	Description
`allModules`	The setting to select all of the modules in a storage array that have audible alarms that you want to turn on, turn off, or mute.
`module`	The specific module that has the audible alarm that you want to turn on, turn off, or mute. Module ID values are `0` to 99. Enclose the module ID value in square brackets ([ ]).
`alarm`	The setting for the audible alarm. This alarm has these values: ■ `enable` – The audible alarm is turned on and sounds if a fault occurs. ■ `disable` – The audible alarm is turned off and does not sound if a fault occurs. ■ `mute` – The audible alarm is turned off if it is sounding. (If another fault occurs after you set the audible alarm to `mute`, the audible alarm sounds again.)

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
```

Parameters

Parameter	Description
`module`	The serial number of the controller module, array module, or the drive module for which you are setting the module ID. Serial numbers can be any combination of alphanumeric characters and any length. Enclose the serial number in double quotation marks (" ").
`id`	The value for the controller module module ID, array module module ID, or the drive module module ID. Module ID values are `0` through 99. You do not need to enclose the module ID value in parentheses.

Notes

This command originally supported the 6540 controller module. The 6540-series controller modules can connect to a variety of drive modules, including those whose module IDs are set by switches. When connecting a 6540-series controller module to drive modules whose module IDs are set by switches, valid values for module IDs for the controller module are `80` through 99. This range avoids conflicts with module IDs that are used for attached drive modules.

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)
```

Parameters

Parameter	Description
`module`	The module where the power-fan CRU, the interconnect CRU, the ESM CRU, or the battery CRU resides. Module ID values are `0` to 99. Enclose the module ID value in square brackets ([ ]). If you do not enter a module ID value, the module ID of the controller module is the default value.
`powerFan`	The Service Action Allowed indicator light on the power-fan CRU that you want to turn on or turn off. Valid power-fan CRU identifiers are `left`, `right`, `top`, or `bottom`. Enclose the power-fan CRU identifier in square brackets ([ ]).
`interconnect`	The Service Action Allowed indicator light for the interconnect-battery CRU.
`esm`	The Service Action Allowed indicator light for an ESM CRU. Valid ESM CRU identifiers are `left`, `right`, `top`, or `bottom`.
`battery`	The Service Action Allowed indicator light for a battery. Valid battery identifiers are `left` or `right`.
`serviceAllowedIndicator`	The setting to turn on or turn off the Service Action Allowed indicator light. To turn on the Service Action Allowed indicator light, set this parameter to `on`. To turn off the Service Action Allowed indicator light, set this parameter to `off`.

Example

This command turns on the Service Action Allowed indicator light for the left ESM in module 5 with the IP address of 155.155.155.155.

```
SMcli 123.145.167.214 123.145.167.215 -c "set module [5]
ESM [left] serviceAllowedIndicator=on;"
```

Notes

This command was originally defined for use with the 6540 controller module. This command is not supported by controller modules that were shipped before the introduction of the 6540 controller module.

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)
```

Parameters

Parameter	Description
pool	The alphanumeric identifier (including - and _) of the pool for which you want to set properties. Enclose the pool identifier in square brackets ([ ]).
addDrives	The location of the drive that you want to add to the pool. Specify the module ID value and the slot ID value for the drive. Module ID values are 0 to 99. Slot ID values are 1 to 32. Enclose the module ID value and the slot ID value in parentheses.
raidLevel	The RAID level for the pool. Valid values are 0, 1, 3, 5, or 6.
owner	The controller that owns the pool. Valid controller identifiers are a or b, where a is the controller in slot A, and b is the controller in slot B. Use this parameter only if you want to change the pool owner.

Notes

Host I/O errors might result in pools with more than 32 volumes. This operation also might result in internal controller reboots because the timeout period ends before the pool definition is set. If you experience this issue, quiesce the host I/O operations, and try the command again.

When you use this command, you can specify one or more of the parameters.

NOTE Specifying the addDrives parameter or the raidLevel parameter starts a long-running operation that you cannot stop.

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
```

Parameter

Parameter	Description
pool	The alphanumeric identifier (including - and _) of the pool that you want to place in a Forced state. Enclose the pool identifier in square brackets ([ ]).

Notes

You can move the drives that comprise a pool from one storage array to another storage array. The CLI provides three commands that let you move the drives. The commands are `start pool export`, `start pool import`, and `set pool forcedState`.

In the Forced state, you can perform an import operation on the pool.

Set Remote Mirror

This command defines the properties for a remote-mirrored 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)
```

Parameters

Parameter	Description
`localVolume` or `localVolumes`	The name of the primary volume for which you want to define properties. You can enter more than one primary volume name. Enclose the primary volume name in square brackets ([ ]). If the primary volume name has special characters, you also must enclose the primary volume name in double quotation marks (" ").
`role`	The setting for the volume to act as the primary volume or the secondary volume. To define the volume as the primary volume, set this parameter to `primary`. To define the volume as the secondary volume, set this parameter to `secondary`. This parameter applies only when the volume is part of a mirror relationship.
`force`	The role reversal is forced if the communications link between the storage arrays is down and promotion or demotion on the local side results in a dual-primary condition or a dual-secondary condition. To force a role reversal, set this parameter to TRUE. The default value is FALSE.
`syncPriority`	The priority that full synchronization has relative to host I/O activity. Valid values are `highest`, `high`, `medium`, `low`, or `lowest`.

Parameter	Description
autoResync	The settings for automatic resynchronization between the primary volumes and the secondary volumes of a remote-mirrored pair. This parameter has these values: ■ enabled – Automatic resynchronization is turned on. You do not need to do anything further to resynchronize the primary volume and the secondary volume. ■ disabled – Automatic resynchronization is turned off. To resynchronize the primary volumes and the secondary volume, you must run the resume remoteMirror command.
writeOrder	This parameter defines write order for data transmission between the primary volume and the secondary volume. Valid values are preserved or notPreserved.
writeMode	This parameter defines how the primary volume writes to the secondary volume. Valid values are synchronous or asynchronous.

Notes

When you use this command, you can specify one or more of the optional parameters.

Synchronization priority defines the amount of system resources that are used to synchronize the data between the primary volumes and the secondary volumes of a mirror relationship. If you select the highest priority level, the data synchronization uses the most system resources to perform the full synchronization, which decreases the performance for host data transfers.

The writeOrder parameter applies only to asynchronous mirrors and makes them become part of a consistency group. Setting the writeOrder parameter to preserved causes the remote-mirrored pair to transmit data from the primary volume to the secondary volume in the same order as the host writes to the primary volume. In the event of a transmission link failure, the data is buffered until a full synchronization can occur. This action can require additional system overhead to maintain the buffered data, which slows operations. Setting the writeOrderparameter to notPreserved frees the system from having to maintain data in a buffer, but it requires forcing a full synchronization to make sure that the secondary volume has the same data as the primary volume.

Set Session

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

Syntax

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

Parameters

Parameter	Description
`errorAction`	How the session responds if an error is encountered during processing. You can choose to stop the session if an error is encountered, or you can continue the session after encountering an error. The default value is `stop`. (This parameter defines the action for execution errors, not syntax errors. Some error conditions might override the `continue` value.)
`password`	The password for the storage array. Enclose the password in double quotation marks (" ").
`performanceMonitorInterval`	The frequency of gathering performance data. Enter an integer value for the polling interval, in seconds, for which you want to capture data. The range of values is 3 to `3600` seconds. The default value is 5 seconds.
`performanceMonitorIterations`	The number of samples to capture. Enter an integer value. The range of values for samples captured is `1` to `3600`. The default value is 5.

Notes

When you use this command, you can specify one or more of the optional parameters.

Passwords are stored on each storage array in a management domain. If a password was not previously set, you do not need a password. The password can be any combination of alphanumeric characters with a maximum of 30 characters. (You can define a storage array password by using the `set storageArray` command.)

The polling interval and the number of iterations that you specify remain in effect until you end the session. After you end the session, the polling interval and the number of iterations return to their default values.

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)
rollbackPriority=(0 | 1 | 2 | 3 | 4)
```

Parameters

Parameter	Description
`volume` or `volumes`	The name of the specific snapshot volume for which you want to define properties. (You can enter more than one volume name if you use the `volumes` parameter). Enclose the snapshot volume name in double quotation marks (" ") inside of square brackets ([ ]).
`userLabel`	A new name that you want to give to a snapshot volume. Enclose the new snapshot volume name in double quotation marks (" ").
`warningThresholdPercent`	The percentage of repository capacity at which you receive a warning that the snapshot repository is nearing full. Use integer values. For example, a value of `70` means 70 percent. The default value is `50`.
`repositoryFullPolicy`	How you want snapshot processing to continue if the snapshot repository volume is full. You can choose to fail writes to the base volume (`failBaseWrites`) or fail writes to the snapshot volume (`failSnapshot`). The default value is `failSnapshot`.
`enableSchedule`	Use this parameter to turn on or to turn off the ability to schedule a snapshot operation. To turn on snapshot scheduling, set this parameter to TRUE. To turn off snapshot scheduling, set this parameter to FALSE.
`schedule`	Use this parameter to schedule a snapshot operation. You can use one of these options for setting a schedule for a snapshot operation: ■ `immediate` ■ `startDate` ■ `scheduleDay` ■ `startTime` ■ `scheduleInterval` ■ `endDate` ■ `noEndDate` ■ `timesPerDay` See the Notes section for information explaining how to use these options.
`rollbackPriority`	Use this parameter to determine whether system resources should be allocated to the rollback operation at the expense of system performance. A value of `0` indicates that the rollback operation is prioritized over all other host I/O. A value of 4 indicates that the rollback operation should be performed with minimal impact to host I/O.

Notes

When you use this command, you can specify one or more of the optional parameters.

You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Names can have a maximum of 30 characters.

You can set the `warningThresholdPercent` parameter and the `repositoryFullPolicy` parameter for both the snapshot repository volume or the snapshot volume.

Scheduling Snapshots

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

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

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

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

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

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))
```

Parameters

Parameter	Description
`alarm`	The setting for the audible alarm. This parameter has these values: ■ `enable` – The audible alarm is turned on and sounds if a fault occurs. ■ `disable` – The audible alarm is turned off and does not sound if a fault occurs. ■ `mute` – The audible alarm is turned off if it is sounding. If another fault occurs after you set the audible alarm to `mute`, the audible alarm sounds again.
`autoSupportConfig`	The setting for automatically collecting support data each time the firmware detects a critical MEL event. This parameter has these values: ■ `enable` – Turns on the collection of support data ■ `disable` – Turns off the collection of support data
`cacheBlockSize`	The cache block size that is used by the controller for managing the cache. Valid values are `4` (4 KB), `8` (8 KB), `16` (16 KB), or `32` (32 KB).
`cacheFlushStart`	The percentage of unwritten data in the cache that causes a cache flush. Use integer values from `0` to `100` to define the percentage. The default value is `80`.
`cacheFlushStop`	The percentage of unwritten data in the cache that stops a cache flush in progress. Use integer values from `0` to `100` to define the percentage. This value must be less than the value of the `cacheFlushStart` parameter.
`defaultHostType`	The default host type of any unconfigured host port to which the controllers are connected. To generate a list of valid host types for the storage array, run the `show storageArray hostTypeTable` command. Host types are identified by a name or a numerical index. Enclose the host type name in double quotation marks (" "). Do not enclose the host type numerical identifier in double quotation marks.
`failoverAlertDelay`	The failover alert delay time in minutes. The valid values for the delay time are `0` to `60` minutes. The default value is `5`.
`mediaScanRate`	The number of days over which the media scan runs. Valid values are `disabled`, which turns off the media scan, or `1` day to `30` days, where `1` day is the fastest scan rate, and `30` days is the slowest scan rate. A value other than `disabled` or `1` to `30` does not allow the media scan to function.

Parameter	Description
`password`	The password for the storage array. Enclose the password in double quotation marks (" ").
`userLabel`	The name for the storage array. Enclose the storage array name in double quotation marks (" ").
`isnsRegistration`	The means of listing the iSCSI target on the iSNS server. Set the parameter to TRUE to list it.

Notes

When you use this command, you can specify one or more of the optional parameters.

Auto Support Data

When enabled, the `set storageArray autoSupportConfig` command causes all configuration and state information for the storage array to be returned each time a critical Major Event Log (MEL) event is detected. The configuration and state information is returned in the form of an object graph. The object graph contains all relevant logical and physical objects and their associated state information for the storage array.

The `set storageArray autoSupportConfig` command collects configuration and state information in this way:

- Automatic collection of the configuration and state information occurs every 72 hours. The configuration and state information is saved to the storage array zip archive file. The archive file has a time stamp that is used to manage the archive files.
- Two storage array zip archive files are maintained for each storage array. The zip archive files are kept on a drive. After the 72-hour time period is exceeded, the oldest archive file is always overwritten during the new cycle.
- After you enable automatic collection of the configuration and state information using this command, an initial collection of information starts. Collecting information after the you issue the command makes sure that one archive file is available and starts the time stamp cycle.

You can run the `set storageArray autoSupportConfig` command on more than one storage array.

Cache Block Size

When you define cache block sizes, use the 4-KB cache block size for storage arrays that require I/O streams that are typically small and random. Use the 8-KB cache block size when the majority of your I/O streams are larger than 4 KB but smaller than 8 KB. Use the 16-KB cache block size or the 32-KB cache block size for storage arrays that require large data transfer, sequential, or high-bandwidth applications.

The `cacheBlockSize` parameter defines the supported cache block size for all of the volumes in the storage array. Not all controller types support all cache block sizes. For redundant configurations, this parameter includes all of the volumes that are owned by both controllers within the storage array.

Cache Flush Start and Cache Flush Stop

When you define values to start a cache flush, a value that is too low increases the chance that data needed for a host read is not in the cache. A low value also increases the number of drive writes that are necessary to maintain the cache level, which increases system overhead and decreases performance.

When setting storage array cache settings, the value of the `cacheFlushStart` parameter must always be greater than or equal to the value of the `cacheFlushStop` parameter. For example, if the value of the `cacheFlushStart` parameter is set to `80`, you may set the value of the `cacheFlushStop` parameter within the range of `0` to `80`.

When you define values to stop a cache flush, the lower the value, the higher the chance that the data for a host read requires a drive read rather than reading from the cache.

Default Host Type

When you define host types, if Storage Domains is enabled, the default host type affects only those volumes that are mapped in the default group. If Storage Domains is not enabled, all of the hosts that are attached to the storage array must run the same operating system and be compatible with the default host type.

Media Scan Rate

Media scan runs on all of the volumes in the storage array that have Optimal status, do not have modification operations in progress, and have the `mediaScanRate` parameter enabled. Use the `set volume` command to enable or disable the `mediaScanRate` parameter.

Password

Passwords are stored on each storage array. For best protection, the password must meet these criteria:

- The password must be between eight and 32 characters long.
- The password must contain at least one uppercase letter.
- The password must contain at least one lowercase letter.
- The password must contain at least one number.
- The password must contain at least one non-alphanumeric character, for example, < > @ +.

NOTE If you are using Encryption Services drives in your storage array, you must use these criteria for your storage array password.

NOTE You must set a password for your storage array before you can create a security key for encrypted Encryption Services drives.

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)
```

Parameter

Parameter	Description
`icmpPingResponse`	This parameter turns on or turns off Echo Request messages. Set the parameter to TRUE to turn on Echo Request messages. Set the parameter to FALSE to turn off Echo Request messages.

Notes

The Internet Control Message Protocol (ICMP) is used by operating systems in a network to send error messages, test packets, and informational messages related to the IP, such as a requested service is not available or that a host or router could not be reached. The ICMP response command sends ICMP Echo Request messages and receives ICMP Echo Response messages to determine if a host is reachable and the time it takes for packets to get to and from that host.

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
```

Parameters

Parameters	Description
`isnsIPv4ConfigurationMethod`	The method that you want to use to define the iSNS server configuration. You can enter the IP address for the IPv4 iSNS servers by selecting `static`. For IPv4, you can choose to have a Dynamic Host Configuration Protocol (DHCP) server select the iSNS server IP address by entering `dhcp`. To enable DCHP, you must set the `isnsIPv4Address` parameter to `0.0.0.0`.
`isnsIPv4Address`	The IP address that you want to use for the iSNS server. Use this parameter with the `static` value for IPv4 configurations. If you choose to have a DHCP server set the IP address for an IPv4 Internet iSNS server, you must set the `isnsIPv4Address` parameter to `0.0.0.0`.

Notes

The iSNS protocol facilitates the automated discovery, management, and configuration of iSCSI devices and Fibre Channel devices on a TCP/IP network. iSNS provides intelligent storage discovery and management services comparable to those found in Fibre Channel networks, which allow a commodity IP network to function in a similar capacity as a storage area network. iSNS also facilitates a seamless integration of IP networks and Fibre Channel networks, due to its ability to emulate Fibre Channel fabric services and manage both iSCSI devices and Fibre Channel devices.

The DHCP server passes configuration parameters, such as network addresses, to IP nodes. DHCP enables a client to acquire all of the IP configuration parameters that it needs to operate. DHCP lets you automatically allocate reusable network addresses.

Set Storage Array iSNS Server IPv6 Address

This command sets the IPv6 address for the iSNS server.

Syntax

```
set storageArray isnsIPv6Address=ipAddress
```

Parameter

Parameters	Description
`isnsIPv6Address`	The IPv6 address that you want to use for the iSNS server.

Notes

The iSNS protocol facilitates the automated discovery, management, and configuration of iSCSI devices and Fibre Channel devices on a TCP/IP network. iSNS provides intelligent storage discovery and management services comparable to those found in Fibre Channel networks, which permits a commodity IP network to function in a similar capacity as a storage area network. iSNS also facilitates a seamless integration of IP networks and Fibre Channel networks, due to its ability to emulate Fibre Channel fabric services, and manage both iSCSI devices and Fibre Channel devices. iSNS provides value in any storage network that has iSCSI devices, Fibre Channel devices, or any combination.

Set Storage Array iSNS Server Listening Port

This command sets the iSNS server listening port.

Syntax

```
set storageArray isnsListeningPort=listeningPortIPAddress
```

Parameter

Parameter	Description
`isnsListeningPort`	The IP address that you want to use for the iSNS server listening port. The range of values for the listening port is `49152` to `65535`. The default value is `3205`.

Notes

A listening port resides on the database server and is responsible for these activities:

- Listening (monitoring) for incoming client connection requests
- Managing the traffic to the server

When a client requests a network session with a server, a listener receives the actual request. If the client information matches the listener information, then the listener grants a connection to the database server.

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
```

Parameters

None.

Notes

If the DHCP server is not operating at full capability, or if the DHCP server is unresponsive, the refresh operation can take between two and three minutes to complete.

The `set storageArray isnsServerRefresh` command returns an error if you did not set the configuration method to DHCP. To set the configuration method to DHCP, use the `set storageArray isnsIPV4ConfigurationMethod` command.

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
```

Parameters

Parameter	Description
`daysToNextLearnCycle`	Valid values are `0` through `7`, where `0` is immediately and `7` is in seven days. The `daysToNextLearnCycle` parameter takes place up to seven days after the next scheduled learn cycle.
`day`	Valid values for the `day` parameter include the days of the week (`Sunday`, `Monday`, `Tuesday`, `Wednesday`, `Thursday`, `Friday`, and `Saturday`). Setting the day causes the next learn cycle to be scheduled on the specified day, after the currently scheduled learn cycle.
`time`	The time in 24-hour format; for example 8:00 a.m. is entered as `08:00`. Nine o'clock p.m. is entered as `21:00`, and 9:30 p.m. is entered as `21:30`.

Notes

You can set the learn cycle to occur only once during a seven-day period.

The `time` parameter selects a specific time that you want to run the learn cycle. If a value is not entered, the command uses a default value of `00:00` (midnight).

If the day and time specified are in the past, the next learn cycle takes place on the next possible day specified.

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)
```

Parameter

Parameter	Description
`modulePositions`	A list of all of the module IDs. The sequence of the module IDs in the list defines the positions for the controller module and the drive modules in a storage array. Valid values are `0` to `99`. Enter the module ID values separated with a space. Enclose the list of module ID values in parentheses. For storage arrays where the controller module has a predefined identifier that is not in the range of valid module position values, use the `controller` value.

Notes

This command defines the position of a module in a storage array by the position of the module ID in the `modulePositions` list. For example, if you have a controller module with an ID set to 84 and drive modules with IDs set to 1, 12, and 50, the `modulePositions` sequence (`84 1 12 50`) places the controller module in the first position, drive module 1 in the second position, drive module 12 in the third position, and drive module 50 in the fourth position. The `modulePositions` sequence (`1 84 50 12`) places the controller module in the second position, drive module 1 in the first position, drive module 50 in the third position, and drive module 12 in the fourth position.

NOTE You must include all of the modules in the storage array in the list defined by the `modulePositions` parameter. If the number of modules in the list does not match the total number of modules in the storage array, an error message appears.

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)
```

Parameter

Parameter	Description
`redundancyMode`	Use `simplex` mode when you have a single controller. Use `duplex` mode when you have two controllers.

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])
```

Parameters

Parameter	Description
`PACProxy`	The URL for the location of a proxy auto-config (PAC) file. The file defines the proxy server to be used for remote status notification.
`proxyHost`	The URL for a host that is to be used for remote status notification.
`proxyPort`	The port number of a host that is to be used for remote status notification.

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
```

Parameters

None.

Notes

Security-capable drives have hardware to accelerate cryptographic processing and each has a unique drive key. A security-capable drive behaves like any other drive until it is added to a secured pool, at which time the security-capable drive becomes security-enabled.

Whenever a security-enabled drive is powered on, it requires the correct security key from the controller before it can read or write data. So, a security-enabled drive uses two keys: the drive key that encrypts and decrypts the data and the security key that authorizes the encryption and decryption processes. The `set storageArray securityKey` command commits the security key to all of the controllers and security-enabled drives in the storage array. The Encryption Services feature ensures that if a security-enabled drive is physically removed from a storage array, its data cannot be read by any other device unless the security key is known.

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
```

Parameters

None.

Set Storage Array Unnamed Discovery Session

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

Syntax

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

Parameter

Parameter	Description
`unnamedDiscoverySession`	This parameter turns on or turns off unnamed discovery sessions. Set the parameter to TRUE to turn on unnamed discovery sessions. Set the parameter to FALSE to turn off unnamed discovery sessions.

Notes

Discovery is the process where initiators determine the targets that are available. Discovery occurs at power-on/initialization and also if the bus topology changes, for example, if an extra device is added.

An unnamed discovery session is a discovery session that is established without specifying a target ID in the login request. For unnamed discovery sessions, neither the target ID nor the target portal group ID are available to the targets.

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)
```

Parameters

Parameter	Description
`allVolumes`	The properties for all volumes in the storage array.
`volume` or `volumes`	The name of the specific volume for which you want to define properties. (You can enter more than one volume name if you use the `volumes` parameter.) Enclose the volume name in double quotation marks (" ") inside of square brackets ([ ]).
`volume`	The World Wide Identifier (WWID) of the volume for which you are setting properties. You can use the WWID instead of the volume name to identify the volume. Enclose the WWID in angle brackets (< >).
`cacheFlushModifier`	The maximum amount of time that data for the volume stays in cache before the data is flushed to physical storage. Valid values are listed in the Notes section.
`cacheWithoutBatteryEnabled`	The setting to turn on or turn off caching without batteries. To turn on caching without batteries, set this parameter to TRUE. To turn off caching without batteries, set this parameter to FALSE.
`mediaScanEnabled`	The setting to turn on or turn off media scan for the volume. To turn on media scan, set this parameter to TRUE. To turn off media scan, set this parameter to FALSE. (If media scan is disabled at the storage array level, this parameter has no effect.)
`mirrorCacheEnabled`	The setting to turn on or turn off the mirror cache. To turn on the mirror cache, set this parameter to TRUE. To turn off the mirror cache, set this parameter to FALSE.
`modificationPriority`	The priority for volume modifications while the storage array is operational. Valid values are `highest`, `high`, `medium`, `low`, or `lowest`.
`owner`	The controller that owns the volume. Valid controller identifiers are `a` or `b`, where `a` is the controller in slot A, and `b` is the controller in slot B. Use this parameter only if you want to change the volume owner.
`preReadRedundancyCheck`	The setting to turn on or turn off preread redundancy checking. Turning on preread redundancy checking verifies the consistency of RAID redundancy data for the stripes containing the read data. Preread redundancy checking is performed on read operations only. To turn on preread redundancy checking, set this parameter to TRUE. To turn off preread redundancy checking, set this parameter to FALSE. **NOTE** Do not use this parameter on non-redundant volumes, such as RAID 0 volumes.
`readCacheEnabled`	The setting to turn on or turn off the read cache. To turn on the read cache, set this parameter to TRUE. To turn off the read cache, set this parameter to FALSE.
`writeCacheEnabled`	The setting to turn on or turn off the write cache. To turn on the write cache, set this parameter to TRUE. To turn off the write cache, set this parameter to FALSE.

Parameter	Description
`cacheReadPrefetch`	The setting to turn on or turn off cache read prefetch. To turn off cache read prefetch, set this parameter to FALSE. To turn on cache read prefetch, set this parameter to TRUE.
`protectionInformationDisabled`	The setting to turn on or turn off T10 protection information for a specific volume. For this parameter to have meaning, your volume must be capable of T10 protection information. This parameter changes a volume from one that supports T10 protection information to a volume that cannot support T10 protection information. To remove T10 protection information from a volume that supports T10 protection information, set this parameter to TRUE. To return a volume to supporting T10 protection information, set this parameter to FALSE. **NOTE** If you remove T10 protection information from a volume, you cannot reset T10 protection information for that volume. To reset T10 protection information for the data on a volume, from which you removed T10 protection information, perform these steps: 1. Remove the data from the volume. 2. Delete the volume. 3. Recreate a new volume with the properties of the deleted volume. 4. Set T10 protection information for the new volume. 5. Move the data to the new volume.
`addCapacity`	The setting to increase the storage size (capacity) of the volume for which you are defining properties. Size is defined in units of `bytes`, KB, `MB`, `GB`, or `TB`. The default value is `bytes`.
`addDrives`	The setting to add new drives to the volume. Specify the module ID value and the slot ID value for the drive. Module ID values are `0` to `99`. Slot ID values are `1` to `32`. Enclose the module ID value and the slot ID value in parentheses. Use this parameter with the `addCapacity` parameter if you need to specify additional drives to accommodate the new size.
`redundancyCheckEnabled`	The setting to turn on or turn off redundancy checking during a media scan. To turn on redundancy checking, set this parameter to TRUE. To turn off redundancy checking, set this parameter to FALSE.
`segmentSize`	The amount of data (in KB) that the controller writes on a single drive in a volume before writing data on the next drive. Valid values are `8`, `16`, `32`, `64`, `128`, `256`, or `512`.
`userLabel`	The new name that you want to give an existing volume. Enclose the new volume name in double quotation marks (" ").
`preReadRedundancyCheck`	The setting to check the consistency of RAID redundancy data on the stripes during read operations. Do not use this operation for non-redundant volumes, for example RAID Level 0. To check redundancy consistency, set this parameter to TRUE. For no stripe checking, set this parameter to FALSE.

Parameter	Description
`accessVolume`	The logical unit number for the access volume. The logical unit number is the only property that you can set for the access volume.
`logicalUnitNumber`	The logical unit number that you want to use to map to a specific host. This parameter also assigns the host to a host group.
`host`	The name of the host to which the volume is mapped. Enclose the host name in double quotation marks (" ").
`hostGroup`	The name of the host group to which the volume is mapped. Enclose the host group name in double quotation marks (" "). `defaultGroup` is the host group that contains the host to which the volume is mapped.

Notes

Host I/O errors might result in pools with more than 32 volumes. This operation might also result in internal controller reboots due to the expiration of the timeout period before the operation completes. If you experience this issue, quiesce host I/O, and try the operation again.

When you use this command, you can specify one or more of the optional parameters.

You can apply these parameters to only one volume at a time:

- `addCapacity`
- `segmentSize`
- `userLabel`
- `logicalUnitNumber`

Add Capacity, Add Drives, and Segment Size

Setting the `addCapacity` parameter, the `addDrives` parameter, or the `segmentSize` parameter starts a long-running operation that you cannot stop. These long-running operations are performed in the background and do not prevent you from running other commands. To show the progress of long-running operations, use the `show volume actionProgress` command.

Access Volume

The access volume is the volume in a SAN environment that is used for in-band communication between the storage management software and the storage array controller. This volume uses a LUN address and consumes 20 MB of storage space that is not available for application data storage. An access volume is required only for in-band managed storage arrays. If you specify the `accessVolume` parameter, the only property you can set is the `logicalUnitNumber` parameter.

Cache Flush Modifier

Valid values for the cache flush modifier are listed in this table.

Value	Description
`Immediate`	Data is flushed as soon as it is placed into the cache.
`250`	Data is flushed after 250 ms.
`500`	Data is flushed after 500 ms.
`750`	Data is flushed after 750 ms.
`1`	Data is flushed after 1 s.

Value	Description
1500	Data is flushed after 1500 ms.
2	Data is flushed after 2 s.
5	Data is flushed after 5 s.
10	Data is flushed after 10 s.
20	Data is flushed after 20 s.
60	Data is flushed after 60 s (1 min.).
120	Data is flushed after 120 s (2 min.).
300	Data is flushed after 300 s (5 min.).
1200	Data is flushed after 1200 s (20 min.).
3600	Data is flushed after 3600 s (1 hr).
Infinite	Data in cache is not subject to any age or time constraints. The data is flushed based on other criteria that are managed by the controller.

Cache Without Battery Enabled

Write caching without batteries enables write caching to continue if the controller batteries are completely discharged, not fully charged, or not present. If you set this parameter to TRUE without an uninterruptible power supply (UPS) or other backup power source, you can lose data if the power to the storage array fails. This parameter has no effect if write caching is disabled.

Modification Priority

Modification priority defines the amount of system resources that are used when modifying volume properties. If you select the highest priority level, the volume modification uses the most system resources, which decreases the performance for host data transfers.

Cache Read Prefetch

The `cacheReadPrefetch` parameter enables the controller to copy additional data blocks into cache while the controller reads and copies data blocks that are requested by the host from the drive into cache. This action increases the chance that a future request for data can be fulfilled from cache. Cache read prefetch is important for multimedia applications that use sequential data transfers. The configuration settings for the storage array that you use determine the number of additional data blocks that the controller reads into cache. Valid values for the `cacheReadPrefetch` parameter are TRUE or FALSE.

Segment Size

The size of a segment determines how many data blocks that the controller writes on a single drive in a volume before writing data on the next drive. Each data block stores 512 bytes of data. A data block is the smallest unit of storage. The size of a segment determines how many data blocks that it contains. For example, an 8-KB segment holds 16 data blocks. A 64-KB segment holds 128 data blocks.

When you enter a value for the segment size, the value is checked against the supported values that are provided by the controller at run time. If the value that you entered is not valid, the controller returns a list of valid values. Using a single drive for a single request leaves other drives available to simultaneously service other requests.

If the volume is in an environment where a single user is transferring large units of data (such as multimedia), performance is maximized when a single data transfer request is serviced with a single data stripe. (A data stripe is the segment size that is multiplied by the number of drives in the pool that are used for data transfers.) In this case, multiple drives are used for the same request, but each drive is accessed only once.

For optimal performance in a multiuser database or file system storage environment, set your segment size to minimize the number of drives that are required to satisfy a data transfer request.

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)
```

Parameters

Parameter	Description
`target`	The name of the target volume for which you want to define properties. Enclose the target volume name in square brackets ([ ]). If the target volume name has special characters, you also must enclose the target volume name in double quotation marks (" ").
`source`	The name of the source volume for which you want to define properties. Enclose the source volume name in square brackets ([ ]). If the source volume name has special characters, you also must enclose the source volume name in double quotation marks (" ").
`copyPriority`	The priority that the volume copy has relative to host I/O activity. Valid values are `highest`, `high`, `medium`, `low`, or `lowest`.
`targetReadOnlyEnabled`	The setting so that you can write to the target volume or only read from the target volume. To write to the target volume, set this parameter to FALSE. To prevent writing to the target volume, set this parameter to TRUE.
`copyType`	Use this parameter to identify that a volume copy has a snapshot. If the volume copy has a snapshot, set this parameter to `online`. If the volume copy does not have a snapshot, set this parameter to `offline`.

Notes

When you use this command, you can specify one or more of the optional parameters.

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"]
```

Parameters

Parameter	Description
`controller`	The controller that has the cache backup device on which you are running the diagnostic tests. Valid controller identifiers are a or b, where a is the controller in slot A, and b is the controller in slot B. Enclose the controller identifier in square brackets ([ ]). If you do not specify a controller, the storage management software returns a syntax error.
`file`	The name of the file that contains the result of the diagnostic tests. Enclose the file name in double quotation marks (" "). This command does not automatically append a file extension to the file name. You must add an extension when you enter the file name.

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"
```

Parameters

Parameter	Description
`controller`	The controller that has the cache memory on which you are running the diagnostic tests. Valid controller identifiers are a or b, where a is the controller in slot A, and b is the controller in slot B. Enclose the controller identifier in square brackets ([ ]).
`file`	The name of the file that contains the result of the diagnostic tests. Enclose the file name in double quotation marks (" "). This command does not automatically append a file extension to the file name. You must add an extension when you enter the file name.

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]
```

Parameters

Parameter	Description
`allControllers`	The setting to return information about both controllers in the storage array.
`controller`	The setting to return information about a specific controller in the storage array. Valid controller identifiers are a or b, where a is the controller in slot A, and b is the controller in slot B. Enclose the controller identifier in square brackets ([ ]).
`summary`	The setting to return a concise list of information about both controllers in the storage array.

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]
```

Parameters

Parameter	Description
`controller`	The setting to return information about a specific controller in the storage array. Valid controller identifiers are a or b, where a is the controller in slot A, and b is the controller in slot B. Enclose the controller identifier in square brackets ([ ]).
`file`	The name of the file that contains the results of the diagnostic tests. This command does not automatically append a file extension to the file name. You must add an extension when you enter the file name.

Show Controller NVSRAM

This command returns a list of the NVSRAM byte values for the specified host type. If you do not enter the optional parameters, this command returns a list of all of the NVSRAM byte values.

Syntax

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

Parameters

Parameter	Description
`allControllers`	The setting to return information about both controllers in the storage array.
`controller`	The setting to return information about a specific controller in the storage array. Valid controller identifiers are a or b, where a is the controller in slot A, and b is the controller in slot B. Enclose the controller identifier in square brackets ([ ]).
`hostType`	The index label or number of the host type. Use the `show storageArray hostTypeTable` command to generate a list of available host type identifiers.
`host`	The name of the host that is connected to the controllers. Enclose the host name in double quotation marks (" ").

Notes

Use the `show controller NVSRAM` command to show parts of or all of the NVSRAM before using the `set controller` command to change the NVSRAM values. Before making any changes to the NVSRAM, contact your Sun Customer Care Center representative to learn what regions of the NVSRAM you can modify.

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
```

Parameters

Parameter	Description
`iscsiInitiator`	The name of the iSCSI initiator for which you want to obtain session information. Enclose the iSCSI initiator name in double quotation marks (" "). You must also enclose the name in either square brackets ([ ]) or angle brackets (< >).
`iscsiTarget`	The name of the iSCSI target for which you want to obtain session information. Enclose the iSCSI target name in double quotation marks (" "). You must also enclose the name in either square brackets ([ ]) or angle brackets (< >).

Notes

If you enter this command without defining any arguments, this command returns information about all of the iSCSI sessions that are currently running. The following command returns information about all of the current iSCSI sessions:

```
show iscsiSessions
```

To limit the information that is returned, enter a specific iSCSI initiator or a specific iSCSI target. This command then returns information about the session for only the iSCSI initiator or the iSCSI target that you named.

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"
```

Parameters

Parameter	Description
`primary`	The name of the local volume that you want for the primary volume in the remote-mirrored pair. Enclose the primary volume name in double quotation marks (" ").
`remoteStorageArrayName`	The remote storage array that contains possible volumes for a secondary volume. If the remote storage array name has special characters, you must also enclose the remote storage array name in double quotation marks (" ").

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
```

Parameter

Parameter	Description
`localVolume` or `localVolumes`	The name of the primary volume of the remote mirrored pair for which you want to check synchronization progress. Enclose the primary volume name in double quotation marks (" ") inside of square brackets ([ ]).

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
```

Parameters

Parameter	Description
`allDrives`	The setting to return information about all of the drives in the storage array.
`driveMediaType`	The type of drive media for which you want to retrieve information. The following values are valid types of drive media: ■ HDD – Use this option when you have hard drives in the drive module. ■ SSD – Use this option when you have solid state drives in the drive module. ■ `unknown` – Use this option if you are not sure what types of drive media are in the drive module. ■ `allMedia` – Use this option when you want to use all types of drive media that are in the drive module.
`driveType`	The type of drive for which you want to retrieve information. You cannot mix drive types. Valid drive types are : ■ `fibre` ■ SATA ■ SAS If you do not specify a drive type, the command defaults to `fibre`.
`drive` or `drives`	The location of the drive for which you want to retrieve information. Specify the module ID value and the slot ID value for each drive. Module ID values are `0` to 99. Slot ID values are `1` to 32. Enclose the module ID values and the slot ID values in parentheses.
`summary`	The setting to return the status, the capacity, the data transfer rate, the product ID, and the firmware version for the specified drives.

Notes

To determine information about the type and location of all of the drives in the storage array, use the `allDrives` parameter.

To determine the information about the Fibre Channel, SATA, or SAS drives in the storage array, use the `driveType` parameter.

To determine the type of drive in a specific location, use the `drive` parameter, and enter the module ID and the slot ID for the drive.

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
```

Parameters

Parameter	Description
`driveChannel`	The identifier number of a specific drive channel for which you want to show information. Valid drive channel values are 1, 2, 3, 4, 5, 6, 7, or 8. Enclose the drive channel in square brackets ([ ]). Use this parameter when you want to show the statistics for only one drive channel.
`driveChannels`	The identifier numbers of several drive channels for which you want to show information. Valid drive channel values are 1, 2, 3, 4, 5, 6, 7, or 8. Enclose the drive channels in square brackets ([ ]) with the drive channel value separated with a space. Use this parameter when you want to show the statistics for more than one drive channel.
`allDriveChannels`	The identifier that selects all of the drive channels.

Notes

None.

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
```

Parameters

None.

Notes

When all of the firmware downloads have successfully completed, this command returns good status. If any firmware downloads fail, this command shows the firmware download status of each drive that was targeted. This command returns the statuses shown in this table.

Status	Definition
Successful	The downloads completed without errors.
Not Attempted	The downloads did not start.
Partial Download	The downloads are in progress.
Failed	The downloads completed with errors.

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]
```

Parameters

Parameter	Description
`controller`	The controller that has the host interface card on which you are running the diagnostic tests. Valid controller identifiers are a or b, where a is the controller in slot A, and b is the controller in slot B. Enclose the controller identifier in square brackets ([ ]). If you do not specify a controller, the storage management software returns a syntax error.
`progressOnly`	The `progressOnly` parameter, shows the progress of the diagnostic test without waiting for the diagnostic tests to completely finish.
`file`	The name of the file that contains the results of the diagnostic tests. This command does not automatically append a file extension to the file name. You must add an extension when you enter the file name.

Notes

The `progressOnly` parameter is useful for seeing the progress of command scripts that need to sequentially complete operations.

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
```

Parameters

None.

Notes

This command returns HBA host port information similar to this example.

```
HOST PORT IDENTIFIER     HOST PORT NAME  HOST TYPE
12:34:56:54:33:22:22:22  Jupiter1        Solaris
12:34:56:78:98:98:88:88  Pluto1          Windows 2000/Server 2003 Clustered
54:32:12:34:34:55:65:66  Undefined       Undefined
```

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]
```

Parameter

Parameter	Description
pool	The alphanumeric identifier of the pool (including - and _) for which you want to show information. Enclose the pool identifier in square brackets ([ ]).

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
```

Parameter

Parameter	Description
pool	The alphanumeric identifier (including - and _) of the pool for which you want to show export dependencies. Enclose the pool identifier in square brackets ([ ]).

Notes

This command spins up the drives in a pool, reads the DACstore, and shows a list of import dependencies for the pool. The pool must be in an Exported state or a Forced state.

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)]
```

Parameters

Parameter	Description
`pool`	The alphanumeric identifier (including - and _) of the pool for which you want to show import dependencies. Enclose the pool identifier in square brackets ([ ]).
`cancelImport`	The setting to spin the drives back down after the pool dependencies have been read. To spin down the drives, set this parameter to TRUE. To let the drives stay spinning, set this parameter to FALSE.

Notes

This command returns the dependencies of a specific pool, which must be in an Exported state or a Forced state. If a decision is made to retain the listed dependencies, then the `cancelImport` parameter can be enforced to spin the drives back down.

You must run the `show pool importDependencies` command before you run the `start pool import` command.

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
```

Parameters

Parameter	Description
`profile`	The parameter to show all of the properties of the logical components and the physical components that comprise the storage array. The information appears in several screens.
`autoSupportConfig`	The parameter to return information about the current state of the operation to automatically collect support data. The following information is returned: ■ Whether the operation is enabled or disabled ■ The location of the folder where the support data file is located
`batteryAge`	The parameter to show the status, the age of the battery in days, and the number of days until the battery needs to be replaced.
`connections`	The parameter to show a list of where the drive channel ports are located and where the drive channels are connected.
`defaultHostType`	The parameter to show the default host type and the host type index.
`healthStatus`	The parameter to show the health, logical properties, and physical component properties of the storage array.
`hostTypeTable`	The parameter to show a table of all of the host types that are known to the controller. Each row in the table shows a host type index and the platform that the host type index represents.
`hotSpareCoverage`	The parameter to show information about which volumes of the storage array have hot spare coverage and which volumes do not.
`features`	The parameter to show a list of the feature identifiers for all enabled premium features in the storage array.
`time`	The parameter to show the current time to which both controllers in the storage array are set.
`volumeDistribution`	The parameter to show the current controller owner for each volume in the storage array.
`longRunningOperations`	The parameter to show the long running operations for each pool and each volume in the storage array. The `longRunningOperation` parameter returns this information: ■ Name of the pool or volume ■ Long running operation ■ Status ■ % complete ■ Time left
`summary`	The parameter to show a concise list of information about the storage array configuration.

Notes

The `profile` parameter shows detailed information about the storage array. The information appears on several screens on a display monitor. You might need to increase the size of your display buffer to see all of the information. Because this information is so detailed, you might want to save the output of this parameter to a file. To save the output to a file, run the `show storageArray` command that looks like this example.

```
-c "show storageArray profile;" -o "c:\\folder\\storageArrayProfile.txt"
```

The previous command syntax is for a host that is running a Windows operating system. The actual syntax varies depending on your operating system.

The `profile` parameter also returns information about the power supplies if the storage array has that capability.

The `batteryAge` parameter returns information in this form.

```
Battery status: Optimal
    Age: 1 day(s)
    Days until replacement: 718 day(s)
```

The `defaultHostType` parameter returns information in this form.

```
Default host type: Linux (Host type index 6)
```

The `healthStatus` parameter returns information in this form.

```
Storage array health status = optimal.
```

The `hostTypeTable` parameter returns information in this form.

```
NVSRAM HOST TYPE INDEX DEFINITIONS
INDEX  AVT STATUS  TYPE
0    Disabled  Windows NT Non-Clustered (SP5 or higher)
1 (Default)  Disabled Windows 2000/Server 2003 Non-Clustered
2    Disabled  Solaris
3    Enabled   HP-UX
4    Disabled  AIX
5    Disabled  Irix
6    Enabled   Linux
7    Disabled  Windows NT Clustered (SP5 or higher)
8    Disabled  Windows 2000/Server 2003 Clustered
9    Enabled   Netware Non-Failover
10   Enabled   PTX
11   Enabled   Netware Failover
12   Enabled   Solaris (with Veritas DMP)
```

The `hotSpareCoverage` parameter returns information in this form.

```
The following pools are not protected: 2, 1
Total hot spare drives: 0
   Standby: 0
   In use: 0
```

The `features` parameter returns information in this form.

```
storagePartitionMax
snapshot
remoteMirror
volumeCopy
```

The `time` parameter returns information in this form.

```
Controller in Slot A
Date/Time: Thu Jun 03 14:54:55 MDT 2004
Controller in Slot B
Date/Time: Thu Jun 03 14:54:55 MDT 2004
```

The `longRunningOperations` parameter returns information in this form:

```
LOGICAL DEVICES  OPERATION         STATUS        TIME REMAINING
Volume-2         Volume Disk Copy  10% COMPLETED  5 min
```

The information fields returned by the `longRunningOperations` parameter have these meanings:

- NAME is the name of a volume that is currently in a long running operation. The volume name must have the "Volume" as a prefix.
- `OPERATION` lists the operation being performed on the pool or volume.
- `% COMPLETE` shows how much of the long running operation has been performed.
- STATUS can have one of these meaings:
  - Pending – The long running operation has not started but will start after the current operation is completed.
  - In Progress – The long running operation has started and will run until completed or stopped by user request.
- `TIME LEFT` indicates the duration remaining to completing the current long running operation. The time is in an "hours minute" format. If less than an hour remains, only the minutes are shown. If less than a minute remains, the message "`less than a minute`" is shown.

The `volumeDistribution` parameter returns information in this form.

```
volume name: 10
     Current owner is controller in slot: A
volume name: CTL 0 Mirror Repository
     Current owner is controller in slot: A
volume name: Mirror Repository 1
     Current owner is controller in slot:A
volume name: 20
     Current owner is controller in slot:A
volume name: JCG_Remote_MirrorMenuTests
     Current owner is controller in slot:A
```

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)
driveMediaType=(HDD | SSD | unknown | allMedia)
raidLevel=(0 | 1 | 3 | 5 | 6)
poolWidth=numberOfDrives
poolCount=numberOfPools
volumesPerGroupCount=numberOfVolumesPerGroup
hotSpareCount=numberOfHotspares
segmentSize=segmentSizeValue
cacheReadPrefetch=(TRUE | FALSE)
securityType=(none | capable | enabled)]
```

Parameters

Parameter	Description
`driveType`	The type of drives that you want to use for the storage array. The `driveType` parameter is not required if only one type of drive is in the storage array. You must use this parameter when you have more than one type of drive in your storage array. Valid drive types are : ■ `fibre` ■ SATA ■ SAS If you do not specify a drive type, the command defaults to `fibre`.
`driveMediaType`	The type of drive media that you want to use for the mirror repository pool. Valid drive media are these: ■ HDD – Use this option when you have hard drives in the drive module. ■ SSD – Use this option when you have solid state drives in the drive module. ■ `unknown` – Use if you are not sure what types of drive media are in the drive module. ■ `allMedia` – Use this option when you want to use all types of drive media that are in the drive module. Use this parameter when you use the `repositoryDriveCount` parameter. You must use this parameter when you have more than one type of drive media in your storage array.
`raidLevel`	The RAID level of the pool that contains the drives in the storage array. Valid RAID levels are `0`, `1`, 3, 5, or 6.
`poolWidth`	The number of drives in a pool in the storage array, which depends on the capacity of the drives. Use integer values.
`poolCount`	The number of pools in the storage array. Use integer values.
`volumesPerGroupCount`	The number of equal-capacity volumes per pool. Use integer values.
`hotSpareCount`	The number of hot spares that you want in the storage array. Use integer values.
`segmentSize`	The amount of data (in KB) that the controller writes on a single drive in a volume before writing data on the next drive. Valid values are 8, 16, 32, 64, 128, 256, or 512.
`cacheReadPrefetch`	The setting to turn on or turn off cache read prefetch. To turn off cache read prefetch, set this parameter to FALSE. To turn on cache read prefetch, set this parameter to TRUE.
`securityType`	The setting to specify the security level when creating the pools and all associated volumes. These settings are valid: ■ `none` – The pool and volumes are not secure. ■ `capable` – The pool and volumes are capable of having security set, but security has not been enabled. ■ `enabled` – The pool and volumes have security enabled.

Notes

If you do not specify any properties, this command returns the RAID Level 5 candidates for each drive type. If RAID Level 5 candidates are not available, this command returns candidates for RAID Level 6, RAID Level 3, RAID Level 1, or RAID Level 0. When you specify auto configuration properties, the controllers validate that the firmware can support the properties.

Drives and Pool

A pool is a set of drives that are logically grouped together by the controllers in the storage array. The number of drives in a pool is a limitation of the RAID level and the controller firmware. When you create a pool, follow these guidelines:

- Beginning with firmware version 7.10, you can create an empty pool so that you can reserve the capacity for later use.
- You cannot mix drive types, such as SAS and Fibre Channel, within a single pool.
- The maximum number of drives in a pool depends on these conditions:
  - The type of controller
  - The RAID level
- RAID levels include: 0, 1, 10, 3, 5, and 6.
  - In a 6140 or a 6140 storage array, a pool with RAID level 0 and a pool with RAID level 10 can have a maximum of 112 drives.
  - In a 6540 storage array, a pool with RAID level 0 and a pool with RAID level 10 can have a maximum of 224 drives.
  - A pool with RAID level 3, RAID level 5, or RAID level 6 cannot have more than 30 drives.
  - A pool with RAID level 6 must have a minimum of five drives.
  - If a pool with RAID level 1 has four or more drives, the storage management software automatically converts the pool to a RAID level 10, which is RAID level 1 + RAID level 0.
- If a pool contains drives that have different capacities, the overall capacity of the pool is based on the smallest capacity drive.
- To enable module loss protection, you must create a pool that uses drives located in at least three drive modules.

Hot Spares

Hot spare drives can replace any failed drive in the storage array. A hot spare drive must have capacity greater than or equal to any drive that can fail. If a hot spare drive is smaller than a failed drive, you cannot use the hot spare drive to rebuild the data from the failed drive. Hot spare drives are available only for RAID Level 1, RAID Level 3, RAID Level 5, or RAID Level 6.

Segment Size

The size of a segment determines how many data blocks that the controller writes on a single drive in a volume before writing data on the next drive. Each data block stores 512 bytes of data. A data block is the smallest unit of storage. The size of a segment determines how many data blocks that it contains. For example, an 8-KB segment holds 16 data blocks. A 64-KB segment holds 128 data blocks.

When you enter a value for the segment size, the value is checked against the supported values that are provided by the controller at run time. If the value that you entered is not valid, the controller returns a list of valid values. Using a single drive for a single request leaves other drives available to simultaneously service other requests. If the volume is in an environment where a single user is transferring large units of data (such as multimedia), performance is maximized when a single data transfer request is serviced with a single data stripe. (A data stripe is the segment size that is multiplied by the number of drives in the pool that are used for data transfers.) In this case, multiple drives are used for the same request, but each drive is accessed only once.

For optimal performance in a multiuser database or file system storage environment, set your segment size to minimize the number of drives that are required to satisfy a data transfer request.

Cache Read Prefetch

Cache read prefetch lets the controller copy additional data blocks into cache while the controller reads and copies data blocks that are requested by the host from the drive into cache. This action increases the chance that a future request for data can be fulfilled from cache. Cache read prefetch is important for multimedia applications that use sequential data transfers. The configuration settings for the storage array that you use determine the number of additional data blocks that the controller reads into cache. Valid values for the `cacheReadPrefetch` parameter are TRUE or FALSE.

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
```

Parameters

None.

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"]]
```

Parameters

Parameter	Description
`host`	The name of a specific host for which you want to see the LUN mappings. Enclose the host name in double quotation marks (" ") inside of square brackets ([ ]).
`hostGroup`	The name of a specific host group for which you want to see the LUN mappings. Enclose the host group name in double quotation marks (" ") inside of square brackets ([ ]).

Show Storage Array Negotiation Defaults

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

Syntax

```
show storageArray  iscsiNegotiationDefaults
```

Parameters

None.

Notes

Information returned includes RAID controller module default settings (that is, those settings that are the starting point for negotiation) and the current active settings.

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
```

Parameter

None.

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
```

Parameters

None.

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
```

Parameters

None.

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"`

Parameters

None.

Notes

Enclose the string in double quotation marks (" ").

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 mirror 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
```

Parameters

Parameter	Description
`allVolumes`	The setting to return information about all of the volumes in the storage array.
`volume` or `volumes`	The name of the specific volume for which you are retrieving information. You can enter more than one volume name. Enclose the volume name in square brackets ([ ]). If the volume name has special characters, you also must enclose the volume name in double quotation marks (" ").
`summary`	The setting to return a concise list of information about the volumes.

Notes

For snapshot volume copies, the `show volume` command returns information about the schedules for the snapshot volume copies. The schedule information is in this form:

```
Schedule State: “Active” | “Disabled” | “Completed”
Last Run Time: <mm/dd/yyyy> <hh:mm a.m. | p.m.>
Next Run Time: <mm/dd/yyyy> <hh:mm a.m. | p.m.>
Start Date: <mm/dd/yyyy>End Date: <mm/dd/yyyy> | "No End Date"
Days of Week: <Sunday - Saturday>, <Sunday - Saturday>, ….
Times for snapshot recreate: <hh:mm a.m. | p.m.>, <hh:mm a.m. | p.m.>
```

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
```

Parameter

Parameter	Description
`volume`	The name of the volume that is running the long-running operation. Enclose the volume name in double quotation marks (" ") inside of square brackets ([ ]).

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"])
```

Parameters

Parameter	Description
`allVolumes`	The setting to return information about volume copy operations for all of the volume copy pairs.
`source`	The name of the source volume about which you want to retrieve information. Enclose the source volume name in double quotation marks (" ") inside of square brackets ([ ]).
`target`	The name of the target volume about which you want to retrieve information. Enclose the target volume name in double quotation marks (" ") inside of square brackets ([ ]).

Show Volume Copy Source Candidates

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

Syntax

```
show volumeCopy sourceCandidates
```

Parameters

None.

Notes

This command returns volume copy source information as shown in this example.

```
Volume Name: finance
   Capacity: 4.0 GB
   Pool: 1
Volume Name: engineering
   Capacity: 4.0 GB
   Pool: 2
```

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
```

Parameter

Parameter	Description
`source`	The name of the source volume for which you are trying to find a candidate target volume. Enclose the source volume name in double quotation marks (" ") inside of square brackets ([ ]).

Show Volume Performance Statistics

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

Syntax

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

Parameters

Parameter	Description
`allVolumes`	The setting to return performance statistics about all of the volumes in the storage array.
`volume` or `volumes`	The name of the specific volume for which you are retrieving performance statistics. You can enter more than one volume name. Enclose the volume name in square brackets ([ ]). If the volume name has special characters, you also must enclose the volume name in double quotation marks (" ").

Show Volume Reservations

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

Syntax

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

Parameters

Parameter	Description
`allVolumes`	The setting to return persistent reservation information about all of the volumes in the storage array.
`volume` or `volumes`	The name of the specific volume for which you are retrieving persistent reservation information. You can enter more than one volume name. Enclose the volume name in square brackets ([ ]). If the volume name has special characters, you also must enclose the volume name in double quotation marks (" ").

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)]
```

Parameters

Parameter	Description
`cacheBackupDevice`	The identifier for the cache backup device on which you want to run the diagnostic tests. Valid cache backup device identifiers are `1`, `2`, `3`, `4` or `all`. ■ `1` for USB1 on the controller circuit board ■ `2` for USB2 on the controller circuit board ■ `3` for USB3 on the controller circuit board ■ `4` for USB4 on the controller circuit board ■ `all` for all of the USBs on the controller circuit board **NOTE** – If you have only one cache backup device, the `all` identifier does not work. Enclose the identifier for the cache backup device in square brackets ([ ]).
`controller`	The controller that has the cache backup device on which you want to run the diagnostic tests. Valid controller identifiers are `a` or `b`, where `a` is the controller in slot A, and `b` is the controller in slot B. Enclose the controller identifier in square brackets ([ ]). If you do not specify a controller, the storage management software returns a syntax error.
`diagnosticType`	The level of diagnostic testing that you want to run on the cache backup device. You can run one of these levels of testing: `basic` – This option validates the basic operation of the ability of the cache backup device to store cache data. This option determines these capabilities of the cache backup device: ■ Whether the cache backup device is write protected or the cache can write data to the device. ■ If the cache backup device is approaching its write cycle limit. `extended` – This option enables you to run more comprehensive diagnostic tests on the host interface card.
`extendedTestID`	This parameter selects the extended test option that you want to run. If you choose the `extended` parameter, you also must also use the `extendedTestID` parameter and one of the extended test options.

Extended Test Option	Description
writePatterns	This option writes a predefined pattern of data in blocks to the entire cache backup device. Each block that was written is then read back, and the data is verified for integrity and accuracy.
random	This option writes a random pattern to each flash block in the cache backup device.

Notes

- When an unexpected power loss occurs, cache memory can have data that has not been written to the drives. This data must be preserved so that it can be written to the drives when power is restored. The contents of the cache memory are backed up to a persistent storage device, such as a USB flash drive, a SATA drive, or a solid state device (SSD).

- The total storage capacity of the flash drives must be equal to the total cache memory, considering that all storage space in a flash drive is not always usable. For example, in a 1-GB flash drive, approximately 968 MB is usable. Also, in some flash drives, the Cyclic Redundancy Check (CRC) needs to be stored along with the data. Because the metadata region is persisted in these flash drives, the storage capacity for the flash drives must be greater than the size of the cache memory.
- You can run the diagnostic test on only one controller in the storage array at any one time.

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)]
```

Parameters

Parameter	Description
`controller`	The controller that has the cache memory on which you want to run the diagnostic tests. Valid controller identifiers are `a` or `b`, where `a` is the controller in slot A, and `b` is the controller in slot B. Enclose the controller identifier in square brackets ([ ]). If you do not specify a controller, the storage management software returns a syntax error.
`diagnosticType`	The level of diagnostic testing that you want to run on the host interface card. You can run one of these levels of testing: `basic` – This option validates the ability of the cache memory to address and access data. `extended` – This option enables you to run more comprehensive diagnostic tests on the host interface card.
`extendedTestID`	This parameter selects the extended test option that you want to run. If you choose the `extended` parameter, you also must use the `extendedTestID` parameter and one of the extended test options.

Extended Test Option	Description
`marchC`	This option performs a March C test on specific regions of the Reconfigurable Processor Assembly (RPA) memory. This option tests for only one set of inverse patterns.
`patterns`	This option performs a word pattern test where the test sequence proceeds with a series of read/write operations for all locations in the specified memory region. The test uses a set of special patterns. The test writes and verifies several patterns at 32-bit widths.
`pseudoRndm`	This option generates a non-repetitive pattern for double word length, writes the pattern to the entire region, and reads back the pattern for verification.
`DMAcopy`	This option tests the capability of Direct Memory Access (DMA) copy operations across regions in the cache memory. This options uses the RPA hardware capabilities to move the data from one region to another region.

Notes

- You can run the diagnostic test on only one controller in the storage array at any one time.

Start Configuration Database Diagnostic

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

Syntax

```
start storageArray configDbDiagnostic
```

Parameters

None.

Notes

Upon completion of the diagnostic test, the controller firmware returns one of these results:

- `Diagnosis completed without errors. No ZIP file created.`
- `Diagnosis completed with errors. Refer to the ZIP file created at: ...\Install_dir\data\FirmwareUpgradeReports\timestamp_buildNo.zip`

If the diagnostic test detects an inconsistency in the configuration database, the controller firmware performs these actions:

- Returns a description of the inconsistency
- Saves a ZIP file containing raw binary data

The controller firmware saves the ZIP file to this location:

```
...\Install_dir\data\FirmwareUpgradeReports\timestamp_buildNo.zip
```

You can use the binary data to help determine the cause of the problem, or you can send the file containing the binary data to a Sun Customer Care Center representative.

To stop the database configuration diagnostic test, use the `stop storageArray configDbDiagnostic` command.

In addition, you can start the database configuration diagnostic test through the storage management software GUI; however, you cannot stop the database configuration diagnostic test through the storage management software GUI. If you want to stop a running diagnostic test, you must use the `stop storageArray configDbDiagnostic` command.

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)]
```

Parameters

Parameter	Description
`controller`	The controller on which you want to run the diagnostic tests. Valid controller identifiers are a or b, where a is the controller in slot A, and b is the controller in slot B. Enclose the controller identifier in square brackets ([ ]). If you do not specify a controller, the storage management software returns a syntax error.
`diagnostic`	The level of diagnostic testing that you want to run on the host interface card. You can run one of these levels of testing: `basic` – This option validates the ability of the base controller to address and access data. `extended` – This option enables you to run more comprehensive diagnostic tests on the base controller card.
`extendedTestID`	This parameter selects the extended test option that you want to run. If you choose the `extended` parameter, you must also used the `extendedTestID` parameter and one of the extended test options.

Extended Test Option	Description
SRAM	This option tests for address, data, and data retention. The address test attempts to write to specific address offsets. The data test attempts to write several data patterns to the address offsets. The data retention test attempts to write a data pattern and then read the data pattern back after a delay. The purpose of the SRAM option is to find memory parity or error correcting code (ECC) errors.
FIFO	This option tests the active processor chip (APC) first in, first out (FIFO) data transmission of the Zip chip. The APC FIFO channels are tested concurrently by writing and verifying different patterns to each channel.
dataCopy	This option tests the ability of the Zip chip to support data copy operations that can copy data from one area of the Zip SDRAM to another area of the Zip SDRAM. This test is performed on any available section of the Zip chip that is not busy.
RAID5Parity	This option tests the ability of the Zip APC to generate and verify RAID 5 parity data. Data buffers are set up in processor memory and parity is generated in processor memory. Some data buffers are set up in parallel architecture (RPA) memory and parity is generated for the data within the RPA memory. The parity that is generated within processor memory is then compared with the parity in the Zip APC.
RAID6Parity	This option tests the ability of the Zip APC to generate and verify RAID 6 parity data. Data buffers are set up in processor memory and parity is generated in processor memory. Some data buffers are set up in redundant parallel architecture (RPA) memory and parity is generated for the data within the RPA memory. The parity that is generated within processor memory is then compared with the parity in the Zip APC.

Notes

You can run the diagnostic test on only one controller in the storage array at any one time.

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)]
```

Parameters

Parameter	Description
`controller`	The controller for which you want to collect the trace debug information. Valid controller identifiers are `a` or `b`, where `a` is the controller in slot A, and `b` is the controller in slot B. You can also simultaneously collect debug for both controllers by entering `both`. Enclose the controller identifier in square brackets ([ ]). If you do not specify a controller, the storage management software returns a syntax error.
`dataType`	The type of data that you want to collect: ■ `current` – Retrieves the current DQ traces ■ `flushed` – Retrieves all flushed DQ traces ■ `currentFlushed` – Retrieves both the current DQ trace and the flushed DQ trace ■ `all` – Retrieves the current DQ trace, flushed DQ trace, and all platform DQ traces **NOTE** – If `dataType=flushed` and `forceFlush=True`, an error message is returned indicating that only active traces can be flushed to the buffer on retrieval.
`forceFlush`	The setting to move the DQ information in the current buffer to the flushed buffer when the DQ trace information defined by the `dataType` parameter is retrieved. To enable force flush, set this parameter to TRUE. To disable force flush, set this parameter to FALSE. **NOTE** – If `dataType=flushed` and `forceFlush=True`, an error message is returned indicating that only active traces can be flushed to the buffer on retrieval.
`file`	The file path and the file name to which you want to save the DQ trace information. Refer to the Notes section for information about naming the files.

Notes

The DQ trace information is written to a compressed file with an extension of .zip. The file name is a combination of a user-defined file name and the storage array identifier (SAID). A constant of "dq" is also added to the file name. The complete file name has this form:

`user_defined_file_name``-SAID-dq.zip`

The compressed file contains the information listed in this table.

File Name	Directory	Comments
user_provided_file_name-SAID-A.dq	SAID/timestamp/	The DQ trace data retrieved from controller A.
user_provided_file_name-SAID-B.dq	SAID/timestamp/	The DQ trace data retrieved from controller B.
user_provided_file_name-SAID-trace_description.xm.	SAID/timestamp/	The description file in an xml format that describes the DQ file attributes for future data mining.

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)
```

Parameters

Parameter	Description
`driveChannel`	The identifier number of the drive channel that you want to locate. Valid values for the identifier number for the drive channel are 1, 2, 3, 4, 5, 6, 7, or 8. Enclose the drive channel identifier number in square brackets ([ ]).
`controller`	The identifier letter of the controller that you want to test. Valid controller identifier values are a or b, where a is the controller in slot A, and b is the controller in slot B.
`testDevices`	The identifiers of the devices (controllers, environmental services monitor [ESMs], or drives) that you want to test. You can specify `all` or enter the specific identifiers for the devices that you want to diagnose.
`dataPattern`	The method of repeatability that you want to test.
`patternNumber`	The hexadecimal data pattern that you want to use to run the test. This number can be any hexadecimal number between `0000` to FFFF. You must place `0x` in front to indicate a hexadecimal number.
`maxErrorCount`	The number of errors that you want to accept before terminating the test.
`testIterations`	The number of times that you want to repeat the test.
`timeout`	The length of time in minutes that you want to run the test.

Notes

Use the `save driveChannel faultDiagnostics` command and the `stop driveChannel faultDiagnostics` command with the `start driveChannel faultDiagnostics` command. These commands are needed to save diagnostic test results to a file and to stop the diagnostic test.

Examples of valid `patternNumber` entries are `0xA5A5`, `0x3C3C`, `8787`, and `1234`.

You can also stop this command at any time by pressing **Ctrl+C**.

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
```

Parameter

Parameter	Description
`driveChannel`	The identifier number of the drive channel that you want to locate. Valid values for the identifier number for the drive channel are 1, 2, 3, 4, 5, 6, 7, or 8. Enclose the drive channel identifier number in square brackets ([ ]).

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
```

Parameter

Parameter	Description
`drive`	The location of the drive that you want to reconstruct. Specify the module ID value and the slot ID value of the drive that you want to revive. Module ID values are 0 to 99. Slot ID values are 1 to 32. Enclose the module ID value and the slot ID value in square brackets ([ ]).

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
```

Parameter

Parameter	Description
`drive`	The location of the drive that you want to reconstruct. Specify the module ID value and the slot ID value of the drive that you want to revive. Module ID values are 0 to 99. Slot ID values are 1 to 32. Enclose the module ID value and the slot ID value in square brackets ([ ]).

Start Drive Reconstruction

This command starts reconstructing a drive.

Syntax

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

Parameter

Parameter	Description
`drive`	The location of the drive that you want to reconstruct. Specify the module ID value and the slot ID value of the drive that you want to revive. Module ID values are `0` to 99. Slot ID values are `1` to 32. Enclose the module ID value and the slot ID value in square brackets ([ ]).

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)]
```

Parameters

Parameter	Description
`hostCard`	The identifier for host interface card on which you want to run the diagnostic tests. Valid host interface card identifiers are `1`, `2`, `3`, or `4`. The value of the identifier is for the position of the host interface card in the controller module or array module. The position of the host interface card depends on the type of controller module or array module in your storage array. See the Notes for more information about the host interface card identifier and the position of the host interface cards in a controller module. Enclose the controller identifier in square brackets ([ ]).
`controller`	The controller that has the host interface card on which you want to run the diagnostic tests. Valid controller identifiers are `a` or `b`, where `a` is the controller in slot A, and `b` is the controller in slot B. Enclose the controller identifier in square brackets ([ ]). If you do not specify a controller, the storage management software returns a syntax error.

Parameter	Description
`diagnosticType`	The level of diagnostic testing that you want to run on the host interface card. You can run one of these levels of testing: `basic` – This option validates the ability of the host interface card to transport I/O data. This option takes approximately 30 seconds to complete. `extended` – This option enables you to run more comprehensive diagnostic tests on the host interface card.
`extendedTestID`	This parameter selects the extended test option that you want to run. If you choose the `extended` parameter, you also must use the `extendedTestID` parameter and one of the extended test options.

Extended Test Option for Fibre Channel	Description
EDC	This option tests the Error Detection and Correction (EDC) generation, verification, and deletion functionality of the QE4 chip. This option tests all modes of the EDC operation, such as, insert, verify, and delete EDC data.
DMA	This option tests the capability of the QE4 chip to take part in a Direct Memory Access (DMA) operation. The DMA can be internal to the chip or can be performed using the services of the raw pool within the Reconfigurable Processor Assembly (RPA) memory.

Extended Test Option for iSCSI	Description
RAM	This option performs a read/write test for the local RAM, the SRAM, and also performs a checksum test for the NVRAM. This option performs the read/write test for the RAM and SRAM by writing data to the memory, reading back the data, and comparing the read data to the written data.
`internalLoopBack`	This option tests the ability of the physical layer (PHY) to transmit data packets over the physical link. For this test, the PHY is set to an internal loopback mode. Data is then transmitted, received, and compared with the original data. The test is run in two passes: ■ For the first pass, the data is predefined by the firmware. ■ For the second pass, the data is generated externally and then transmitted.

Notes

You can run the diagnostic test on only one controller in the storage array at any one time.

A controller can have either one or two host interface cards.

- If a controller has one host interface card, the value for the position of each host interface card depends on the position of the controller in the controller module. The host interface card in the controller in controller module slot A has a position value of 1. The host interface card in the controller in controller module slot B has a position value of 2.

- If a controller has two host interface cards, the value for the position of each host interface card depends on the position of the host interface card in the controller and the position of the controller in the controller module. In most cases the position of the host interface card is identified with labels such as Host Card 1 and Host Card 2 on each controller. The position value of the host interface cards are listed in this table.

Controller	Host Card Label	Position
A	Host Card 1	1
	Host Card 2	2
B	Host Card 1	3
	Host Card 2	4

You cannot use a loopback connection for the host interface card that you are testing.

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
```

Parameter

Parameter	Description
`controller`	The identifier letter of the controller that has the iSCSI host ports. Valid controller identifier values are a or b, where a is the controller in slot A, and b is the controller in slot B.
`iscsiHostPort`	The identifier of the iSCSI port for which you want to refresh the DHCP parameters. Enclose the iSCSI host port identifier in square brackets ([ ]).

Notes

This operation ends the iSCSI connections for the portal and temporarily brings down the portal.

Start Data Replicator Software Synchronization

This command starts Data Replicator Software synchronization.

Syntax

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

Parameter

Parameter	Description
`primary`	The name of the primary volume for which you want to start synchronization. Enclose the primary volume name in double quotation marks (" ") inside of square brackets ([ ]).

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])
```

Parameters

Parameter	Description
`drive` or `drives`	The module and the slot where the drive resides. Module ID values are `0` to 99. Slot ID values are `1` to 32. Enclose the module ID values and the slot ID values in square brackets ([ ]).

Notes

The controller firmware creates a lock that restricts access to the ES drives. ES drives have a state called Security Capable. When you create a security key, the state is set to Security Enabled, which restricts access to all ES drives that exist within the storage array.

Start Snapshot Rollback

This command starts a rollback operation for one or more snapshot volumes. The content of the base volume changes immediately to match the point-in-time content of the selected snapshot volume. The base volume immediately becomes available for read/write requests after the rollback operation has successfully completed. To stop a snapshot rollback operation, use the `stop rollback volume` command.

The repository volume that is associated with the snapshot continues to track any new changes between the base volume and the snapshot volume that occur after the rollback operation is completed.

Syntax

```
start rollback (volume [snapshotVolumeName |
volumes [snapshotVolumeName1 ... snapshotVolumeNamen])
```

Parameter

Parameter	Description
`volume` or `volumes`	The name of the specific snapshot volume or snapshot volumes for which you want to start a rollback operation. Enclose the snapshot volume name in square brackets ([ ]). If the snapshot volume name has special characters, you must also enclose the snapshot volume name in double quotation marks (" ").).

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
```

Parameter

None.

Notes

If you used the `set storageArray isnsIPv4ConfigurationMethod` command to set the configuration but did not set the configuration to DHCP, running the `start storageArray isnsServerRefresh` returns an error.

Start Storage Array Locate

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

Syntax

```
start storageArray locate
```

Parameters

None.

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
```

Parameter

Parameter	Description
`module`	The module that you want to locate. Module ID values are `0` to `99`. Enclose the module ID value in square brackets ([ ]).

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
```

Parameter

Parameter	Description
`pool`	The alphanumeric identifier of the pool (including - and _) that you want to defragment. Enclose the pool identifier in square brackets ([ ]).

Notes

Host I/O errors might result in the pools with more than 32 volumes. This operation also might result in internal controller reboots because the timeout period ends before the pool definition is set. If you experience this issue, quiesce the host I/O operations, and try the command again.

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`

Parameter

Parameter	Description
`pool`	The alphanumeric identifier of the pool (including - and _) that you want to export. Enclose the pool identifier in square brackets ([ ]).

Notes

When this command is successful, you can run the `start pool import` command to finish moving the pool to a Complete state, which makes the pool available to the new storage array.

If this command is unsuccessful because hardware problems prevented the completion of the export, use the `set pool forceState` command. The `set pool forceState` command lets you use the `start pool import` command to import a pool.

After the pool is in an Exported state or a Forced state, you can remove the drives that comprise the pool from the storage array. You can reinstall the drives in a different storage array.

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
```

Parameter

Parameter	Description
pool	The alphanumeric identifier of the pool (including - and _) that you want to import. Enclose the pool identifier in square brackets ([ ]).

Notes

Higher-level volumes that are specifically related to premium features (Snapshot, Data Replicator Software, Volume Copy, mapping, and persistent reservations) are removed as part of the import operation.

You must run the `show pool importDependencies` command before you run the `start pool import` command.

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
```

Parameter

Parameter	Description
pool	The alphanumeric identifier of the pool (including - and _) for which you want to locate the drives that belong to that pool. Enclose the pool identifier in square brackets ([ ]).

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
```

Parameter

Parameter	Description
`volume`	The name of the volume for which you are starting the formatting. Enclose the volume name in square brackets ([ ]). If the volume name has special characters, you also must enclose the volume name in double quotation marks (" ").

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
```

Parameters

Parameter	Description
`controller`	The controller that has the cache backup device on which you are running the diagnostic tests. Valid controller identifiers are a or b, where a is the controller in slot A, and b is the controller in slot B. Enclose the controller identifier in square brackets ([ ]). If you do not specify a controller, the storage management software returns a syntax error.

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
```

Parameter

Parameter	Description
`controller`	The controller that has the cache memory on which you are running the diagnostic tests. Valid controller identifiers are a or b, where a is the controller in slot A, and b is the controller in slot B. Enclose the controller identifier in square brackets ([ ]). If you do not specify a controller, the storage management software returns a syntax error.

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
```

Parameters

None.

Notes

The controller firmware returns a confirmation that the diagnostic test was cancelled.

In addition, you can start the database configuration diagnostic test through the storage management software GUI; however, you cannot stop the database configuration diagnostic test through the storage management software GUI. If you want to stop a running diagnostic test, you must use the `stop storageArray configDbDiagnostic` command.

If you try to use the `stop storageArray configDbDiagnostic` command after validation of the storage array configuration has finished, you do not receive any message that the validation has finished. This behavior is expected.

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
```

Parameters

Parameter	Description
`controller`	The setting to return information about a specific controller in the storage array. Valid controller identifiers are a or b, where a is the controller in slot A, and b is the controller in slot B. Enclose the controller identifier in square brackets ([ ]). If you do not specify a controller, the storage management software returns a syntax error.

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
```

Parameters

None.

Notes

Use the `start driveChannel faultDiagnostics` command and the `save driveChannel faultDiagnostics` command with the `stop driveChannel faultDiagnostics` command. These commands are needed to start the diagnostic test and save diagnostic test results to a file.

You can also stop the `start driveChannel faultDiagnostics` command at any time by pressing **Ctrl+C**.

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
```

Parameters

None.

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
```

Parameters

None.

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
```

Parameters

Parameter	Description
`controller`	The controller that has the host interface card on which you are running the diagnostic tests. Valid controller identifiers are a or b, where a is the controller in slot A, and b is the controller in slot B. Enclose the controller identifier in square brackets ([ ]). If you do not specify a controller, the storage management software returns a syntax error.

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
```

Parameters

None.

Stop Snapshot

This command stops a copy-on-write operation.

Syntax

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

Parameter

Parameter	Description
`volume` or `volumes`	The name of the specific volume for which you want to stop a copy-on-write operation. You can enter more than one volume name. Enclose the volume names using one of these forms: ■ On a Windows command line: \"*volumeName*\" ■ In a Windows script engine window: ["*volumeName*"] ■ On a Linux command line: \"*volumeName*\" ■ In a Linux script engine window: [\"*volumeName*\"]

Notes

Names can be any combination of alphanumeric characters, underscore (_), hyphen (-), and pound (#). Names can have a maximum of 30 characters.

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

If you do not choose a name for either the snapshot volume or the snapshot repository volume, the storage management software creates a default name by using the base volume name. An example of the snapshot volume name that the controllers might create is, if the base volume name is aaa and does not have a snapshot volume, the default snapshot volume name is aaa-1. If the base volume already has *n*-1 number of snapshot volumes, the default name is aaa-*n*. An example of the snapshot repository volume name that the controller might create is, if the base volume name is aaa and does not have a snapshot repository volume, the default snapshot repository volume name is aaa-R1. If the base volume already has *n*-1 number of snapshot repository volumes, the default name is aaa-R*n*.

Stop Snapshot Rollback

This command stops a snapshot rollback operation that was initiated by the the `start rollback volume` command.

NOTE Canceling a rollback operation leaves the base volume in an indeterminate state with potentially invalid or inconsistent data. The related snapshot volume becomes disabled and unusable.

Syntax

```
stop rollback volume [snapshotVolumeName]
```

Parameter

Parameter	Description
`volume`	The name of the specific snapshot volume for which you want to stop a rollback operation. Enclose the snapshot volume name in square brackets ([ ]). If the snapshot volume name has special characters, you must also enclose the snapshot volume name in double quotation marks (" ").

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
```

Parameters

None.

Stop Storage Array iSCSI Session

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

Syntax

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

Parameter

Parameter	Description
`iscsiSession`	The identifier number of the iSCSI session. Enclose the identifier number of the iSCSI session in square brackets ([ ]).

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
```

Parameters

None.

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
```

Parameters

None.

Stop Volume Copy

This command stops a volume copy operation.

Syntax

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

Parameters

Parameter	Description
`target`	The name of the target volume for which you want to stop a volume copy operation. Enclose the target volume name in square brackets ([ ]). If the target volume name has special characters, you also must enclose the target volume name in double quotation marks (" ").
`source`	The name of the source volume for which you want to stop a volume copy operation. Enclose the source volume name in square brackets ([ ]). If the source volume name has special characters, you also must enclose the source volume name in double quotation marks (" ").

Suspend Remote Mirror

This command suspends a Data Replicator Software operation.

Syntax

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

Parameters

Parameter	Description
`primary` or `primaries`	The name of the volume for which you want to suspend operation. Enclose the volume name in square brackets ([ ]). If the volume name has special characters, you must also enclose the volume name in double quotation marks (" ").
`writeConsistency`	This parameter defines whether the volumes identified in this command are in a write-consistency group or are separate. For the volumes in the same write-consistency group, set this parameter to TRUE. For the volumes that are separate, set this parameter to FALSE.

Notes

If you set the `writeConsistency` parameter to TRUE, the volumes must be in a write-consistency group (or groups). This command suspends all write-consistency groups that contain the volumes. For example, if volumes A, B, and C are in a write-consistency group and they have remote counterparts A′, B′, and C′, the command:

```
suspend remoteMirror volume ["A"] writeConsistency=TRUE
```

suspends A-A′, B-B′, and C-C′. If you have a write-consistency group 1={A, B, C} and write-consistency group 2={D, E, F}, the command:

```
suspend remoteMirror volumes ["A" "D"] writeConsistency=TRUE
```

suspends both write-consistency groups.

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"
```

Parameters

Parameter	Description
`file`	The file path and the file name that has the security key. Enclose file path and the file name in double quotation marks (" "). For example: `file="C:\Program Files\CLI\sup\seckey.slk"` **IMPORTANT** – The file name must have an extension of `.slk`.
`passPhrase`	A character string that encrypts the security key so that you can store the security key in an external file. Enclose the pass phrase in double quotation marks (" ").

Notes

Your pass phrase must meet these criteria:

- The pass phrase must be between eight and 32 characters long.
- The pass phrase must contain at least one uppercase letter.
- The pass phrase must contain at least one lowercase letter.
- The pass phrase must contain at least one number.
- The pass phrase must contain at least one non-alphanumeric character, for example, < > @ +.

NOTE If your pass phrase does not meet these criteria, you will receive an error message.

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 184 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=filenamecontent=(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

Printed in U.S.A.

51334-00A