SGI InfiniteStorage 4000 Series and 5000 Series CLI and Script Commands Quick Reference

(ISSM 10.86)

007-5879-002

April 2013

The information in this document supports the SGI InfiniteStorage 4000 series and 5000 series storage systems (ISSM 10.86). Refer to the table below to match your specific SGI InfiniteStorage product with the model numbers used in this document.

SGI Model #	**NetApp Model**
TP9600H	6091
TP9700F	6091
IS4500F	6091
TP9600F	3994 and 3992
IS4000H	3994
IS350	3992
IS220	1932 1333 DE1300
IS4100	4900
IS-DMODULE16-Z	FC4600
IS-DMODULE60	DE6900
IS4600	7091
IS-DMODULE12 & IS2212 (JBOD)	DE1600
IS-DMODULE24 & IS2224 (JBOD)	DE5600
IS-DMODULE60-SAS	DE6600
IS5012	E2600
IS5024	E2600
IS5060	E2600
IS5512	E5400
IS5524	E5400
IS5560	E5400
IS5600	E5500

Copyright information

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

Trademark information

Table of Contents

Formatting the Commands 1

The command line interface (CLI) is a software application that provides a way to configure and monitor storage arrays. Using the CLI, you can run commands from an operating system prompt, such as the DOS `C:` prompt, a Linux operating system path, or a Solaris operating system path.

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

Structure of a CLI Command

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

- A command wrapper identified by the term `SMcli`
- The storage array identifier
- Terminals that define the operation to be performed
- Script commands

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

All CLI commands have the following structure:

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

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

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

Interactive Mode

If you enter `SMcli` and a storage array name but do not specify CLI parameters, script commands, or a script file, the command line interface runs in interactive mode. Interactive mode lets you run individual commands without prefixing the commands with `SMcli`.

In interactive mode, you can enter a single command, view the results, and enter the next command without typing the complete `SMcli` string. Interactive mode is useful for determining configuration errors and quickly testing configuration changes.

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

CLI Command Wrapper Syntax

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

Convention	Definition
`a | b`	Alternative ("a" or "b")
`italicized-words`	A terminal that needs user input to fulfill a parameter (a response to a variable)
`[ ... ]` (square brackets)	Zero or one occurrence (square brackets are also used as a delimiter for some command parameters)
`{ ... }` (curly braces)	Zero or more occurrences
`(a | b | c)`	Choose only one of the alternatives
`bold`	A terminal that needs a command parameter entered to start an action

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

```
SMcli (-n  storage-system-name | -w wwID)
[-c "command; {command2};"]
[-o outputfile] [-p password] [-R (admin | monitor)]
[-e] [-S] [-quick]
```

```
SMcli (-n storage-system-name   -w wwID)
[-f scriptfile]
[-o outputfile] [-R (admin | monitor)] [-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.

Terminal	Definition
`-f` (lowercase)	Specifies a file name that contains script commands that you want to run on the specified storage array. The `-f` terminal is similar to the `-c` terminal in that both terminals are intended for running script commands. The `-c` terminal runs individual script commands. The `-f` terminal runs a file of script commands. By default, any errors that are encountered when running the script commands in a file are ignored, and the file continues to run. To override this behavior, use the `set session errorAction=stop` command in the script file.
`-g`	Specifies an ASCII file that contains email sender contact information that will be included in all email alert notifications. The CLI assumes that the ASCII file is text only, without delimiters or any expected format. Do not use the `-g` terminal if a `userdata.txt` file exists.
`-h`	Specifies the host name that is running the SNMP agent to which the storage array is connected. Use the `-h` terminal with these terminals: ■ `-a` ■ `-x`
`-I` (uppercase)	Specifies the type of information to be included in the email alert notifications. You can select these values: ■ `eventOnly` – Only the event information is included in the email. ■ `profile` – The event and array profile information is included in the email. You can specify the frequency for the email deliveries using the `-q` terminal.
`-i` (lowercase)	Shows the IP address of the known storage arrays. Use the `-i` terminal with the `-d` terminal. The file 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.

Terminal	Definition
`-o`	Specifies a file name for all output text that is a result of running the script commands. Use the `-o` terminal with these terminals: ■ `-c` ■ `-f` If you do not specify an output file, the output text goes to standard output (stdout). All output from commands that are not script commands is sent to stdout, regardless of whether this terminal is set.
`-p`	Defines the password for the storage array on which you want to run commands. A password is not necessary if a password has not been set on the storage array. If you set a monitor password for the storage array, the use of the `-p` parameter is mandatory. Users cannot run any of the non -destructive commands such as the `show` commands.
`-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.

Terminal	Definition
`-quick`	Reduces the amount of time that is required to run a single-line operation. An example of a single-line operation is the `recreate snapshot volume` command. This terminal reduces time by not running background processes for the duration of the command. Do not use this terminal for operations that involve more than one single-line operation. Extensive use of this command can overrun the controller with more commands than the controller can process, which causes operational failure. Also, status updates and configuration updates that are collected usually from background processes will not be available to the CLI. This terminal causes operations that depend on background information to fail.
`-R` (uppercase)	Defines the user role for the password. The roles can be either: ■ `admin` – The user has privilege to change the storage array configuration. ■ `monitor` – The user has privilege to view the storage array configuration, but cannot make changes. The `-R` parameter is valid only when used with the `–p` parameter, which specifies that a you define a password for a storage array. The `-R` parameter is required only if the dual password feature is enabled on the storage array. The `-R` parameter is not necessary under these conditions: ■ The dual password feature is not enabled on the storage array. ■ Only one admin role is set and the monitor role is not set for the storage array.
`-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.

Terminal	Definition
`-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.

Alert Severities Commands

The CLI provides special commands that enable you to set alert severities, and to send out a test alert to the Windows Event Log and all configured syslog receivers. The alert severities apply to all of the storage arrays in the entire storage system. The commands are SMcli commands that run only from a command line.

Setting Alert Severities

```
SMcli -alertSeverities (severity |
[severity1, ... severityN])
```

The alert severities values that you can set are the following:

- `critical` – Alerts will be sent
- `warning` – Alerts will be sent
- `informational` – Alerts will not be sent
- `debug` – Alerts will be sent

NOTE The debug value is for Technical Support only. Do not attempt to use this value.

You can set one or more alert severities values. If you set more than one alert severities value, enclose all of the values in square brackets ([ ])and separate the values by a comma.

Showing Alert Severities

```
SMcli -alertSeverities
```

This command shows all of the severities for which an alert is sent. This command cannot show information for a specific type of severity.

Sending a Test Alert

```
SMcli -alertTest
```

This command sends out a test alert to the Windows Event Log and all configured syslog receivers.

AutoSupport Bundle Collection Commands

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

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

- ASUP collection

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

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

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

In the CLI, ASUP is a nonconfigurable, set it and forget it feature. Using the CLI commands, you can only turn on or turn off ASUP. Once turned on, ASUP automatically reports configuration, subsystem status, and exceptions in near-real time. Because ASUP speeds up troubleshooting and problem analysis, ASUP is the preferred data collection method to use if available on the storage array.

ASUP Messages

ASUP provides these types of messages:

- Event:
  - Sent when a support event occurs on the managed storage array.
  - Includes system configuration and diagnostic information.
  - Includes minimal extent of system configuration information.
- Daily:
  - Sent at midnight, local time of the host.
  - Provides a current set of system event logs and performance data.
  - Places less burden on payload and transmission on the messages originating from Event ASUP messages.

- Weekly:
  - Sent once every week at times that do not impact storage array operations.
  - Includes configuration and system state information.

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

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

- HTTP or HTTPS upload:
- SMTP notifications

ASUP Commands

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

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

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

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

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

ASUP Log

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

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

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`	Tray ID and slot ID
`replacementDrive`	Tray ID and slot ID
`driveChannel`	Drive channel identifier
`host`	User label
`hostChannel`	Host channel identifier

Object Type	Identifier
`hostGroup`	User label
`hostPort`	User label
`iscsiInitiator`	User label or iSCSI Qualified Name (IQN)
`iscsiTarget`	User label or IQN
`snapshot (legacy)`	Volume user label
`storageArray`	Not applicable
`tray`	Tray ID
`volume`	Volume user label or volume World Wide Identifier (WWID) (`set` command only)
`volumeCopy`	Target volume user label and, optionally, the source volume user label
`volumeGroup`	User label Valid characters are alphanumeric, a hyphen, and an underscore.

Statement data is in the form of:

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

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

Synopsis of the Script Commands

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

Table 2 General Form of the Script Commands

Syntax	Description
`accept` *`object`* {*`statement-data`*}	Perform the pending operation.
`activate` *`object`* {*`statement-data`*}	Sets up the environment so that an operation can take place or performs the operation if the environment is already set up correctly.

Syntax	Description

`autoConfigure storageArray`
{*statement-data*}
	Automatically creates a configuration that is based on the parameters that are specified in the command.

`check` *object* {*statement-data*}	Starts an operation to report on errors in the object, which is a synchronous operation.
`clear` *object* {*statement-data*}	Discards the contents of some attributes of an object. This operation is destructive and cannot be reversed.
`create` *object* {*statement-data*}	Creates an object of the specified type.
`deactivate` *object* {*statement-data*}	Removes the environment for an operation.
`delete` *object*	Deletes a previously created object.
`diagnose` *object* {*statement-data*}	Runs a test and shows the results.
`disable` *object* {*statement-data*}	Prevents a feature from operating.
`download` *object* {*statement-data*}	Transfers data to the storage array or to the hardware that is associated with the storage array.
`enable` *object* {*statement-data*}	Sets a feature to operate.
`load` *object* {*statement-data*}	Transfers data to the storage array or to the hardware that is associated with the storage array. This command is functionally similar to the `download` command.
`recopy` *object* {*statement-data*}	Restarts a volume copy operation by using an existing volume copy pair. You can change the parameters before the operation is restarted.
`recover` *object* {*statement-data*}	Re-creates an object from saved configuration data and the statement parameters. (This command is similar to the `create` command.)
`recreate` *object* {*statement-data*}	Restarts a snapshot (legacy) operation by using an existing snapshot (legacy) volume. You can change the parameters before the operation is restarted.
`remove` *object* {*statement-data*}	Removes a relationship from between objects.
`repair` *object* {*statement-data*}	Repairs errors found by the `check` command.
`replace` *object* {*statement-data*}	The specified object replaces an existing object in the storage array.
`reset` *object* {*statement-data*}	Returns the hardware or an object to an initial state.

Syntax	Description
`resume` *`object`*	Starts a suspended operation. The operation starts where it left off when it was suspended.
`revive` *`object`*	Forces the object from the Failed state to the Optimal state. Use this command only as part of an error recovery procedure.

`save` *`object`*
`{statement-data}`
	Writes information about the object to a file.

`set` *`object`*
`{statement-data}`
	Changes object attributes. All changes are completed when the command returns.

`show` *`object`*
`{statement-data}`
	Shows information about the object.

`start` *`object`*
`{statement-data}`
	Starts an asynchronous operation. You can stop some operations after they have started. You can query the progress of some operations.

`stop` *`object`*
`{statement-data}`
	Stops an asynchronous operation.

`suspend` *`object`*
`{statement-data}`
	Stops an operation. You can then restart the suspended operation, and it continues from the point where it was suspended.

Recurring Syntax Elements

Recurring syntax elements are a general category of parameters and options that you can use in the script commands. Table 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-trayID	(0-99)
slotID	(1-32)
portID	(0-127)
drive-spec	*trayID,slotID* or *trayID,drawerID,slotID* A drive is defined as two or three interger literal values separated by a comma. Low-density trays require two values. High-density trays, those trays that have drawers, require three values.
drive-spec-list	*drive-spec drive-spec*
trayID-list	*trayID* {*trayID*}
esm-spec-list	*esm-spec* {*esm-spec*}
esm-spec	*trayID,* (left | right)
hex-literal	*0xhexadecimal-literal*
volumeGroup-number	*integer-literal*
filename	*string-literal*
error-action	(stop | continue)
drive-channel-identifier (four drive ports per tray)	(1 | 2 | 3 | 4)

Recurring Syntax	Syntax Value
drive-channel-identifier (eight drive ports per tray)	(1 | 2 | 3 | 4 | 5 | 6 | 7 | 8)
drive-channel-identifier-list	*drive-channel-identifier* {*drive-channel-identifier*}
host-channel-identifier (four host ports per tray)	(a1 | a2 | b1 | b2)
host-channel-identifier (eight host ports per tray)	(a1 | a2 | a3 | a4 | b1 | b2 | b3 | b4)
host-channel-identifier (16 host ports per tray)	(a1 | a2 | a3 | a4 | a5 | a6 | a7 | a8 | b1 | b2 | b3 | b4 | b5 | b6 | b7 | b8)
drive-type	(fibre | SATA | SAS)
drive-media-type	(HDD | SSD | unknown| allMedia) *HDD* means hard disk drive. *SSD* means solid state disk.
repository-spec	*instance-based-repository-spec* | *count-based-repository-spec*
instance-based-repository-spec	(repositoryRAIDLevel =*repository-raid-level* repositoryDrives= (*drive-spec-list*) [repositoryVolumeGroupUserLabel =*user-label*] [trayLossProtect=(TRUE | FALSE)[1]]) | [drawerLossProtect=(TRUE | FALSE)[2]]) | (repositoryVolumeGroup=*user-label* [freeCapacityArea=*integer-literal*[3]]) Specify the repositoryRAIDLevel parameter with the repositoryDrives parameter. Do not specify the RAID level or the drives with the volume group. Do not set a value for the trayLossProtect parameter when you specify a volume group.
count-based-repository-spec	repositoryRAIDLevel =*repository-raid-level* repositoryDriveCount=*integer-literal* [repositoryVolumeGroupUserLabel =*user-label*] [driveType=*drive-type*[4]] [trayLossProtect=(TRUE | FALSE)[1]] | [drawerLossProtect=(TRUE | FALSE)[2]] | [dataAssurance=(none | enabled)[5]] |

Recurring Syntax	Syntax Value
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*}

autoconfigure-vols-attr-value-pair	driveType=*drive-type* |
	driveMediaType=*drive-media-type* |
	raidLevel=*raid-level* |
	volumeGroupWidth=*integer-literal* |
	volumeGroupCount=*integer-literal* |
	volumesPerGroupCount=*integer-literal*[6] |
	hotSpareCount=*integer-literal* |
	segmentSize=*segment-size-spec* |
	cacheReadPrefetch=(TRUE | FALSE)
	securityType=(none | capable | enabled)[7] |
	dataAssurance=(none | enabled)[5]

create-volume-copy-attr-value-list	*create-volume-copy-attr-value-pair* {*create-volume-copy-attr-value-pair*}

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

Recurring Syntax	Syntax Value

iscsi-host-port-options	IPv4Address=*ip-address* |
	IPv6LocalAddress=*ipv6-address* |
	IPv6RoutableAddress=*ipv6-address* |
	IPv6RouterAddress=*ipv6-address* |
	enableIPv4=(TRUE | FALSE) |
	enableIPv6=(TRUE | FALSE) |
	enableIPv4Priority=(TRUE | FALSE) |
	enableIPv6Priority=(TRUE | FALSE) |
	IPv4ConfigurationMethod=
	(static | dhcp) |
	IPv6ConfigurationMethod=
	(static | auto) |
	IPv4GatewayIP=*ip-address* |
	IPv6HopLimit=*integer* |
	IPv6NdDetectDuplicateAddress=*integer* |
	IPv6NdReachableTime=*time-interval* |
	IPv6NdRetransmitTime=*time-interval* |
	IPv6NdTimeOut=*time-interval* |
	IPv4Priority=*integer* |
	IPv6Priority=*integer* |
	IPv4SubnetMask=*ip-address* |
	IPv4VlanId=*integer* |
	IPv6VlanId=*integer* |
	maxFramePayload=*integer* |
	tcpListeningPort=*tcp-port-id* |
	portSpeed=(autoNegotiate | 1 | 10)

test-devices-list	*test-devices* {*test-devices*}

test-devices	controller=(a | b)
	esms=(*esm-spec-list*)
	drives=(*drive-spec-list*)

snapshot (legacy)-schedule-attribute-value-list	*snapshot (legacy)-schedule-attribute-value-pair* {*snapshot (legacy)-schedule-attribute-value-pair*}
time-zone-spec	(GMT+HH:MM | GMT-HH:MM) [dayLightSaving=HH:MM]

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

[1] For tray loss protection to work, each drive in a volume group must be in a separate tray. If you set the `trayLossProtect` parameter to `TRUE` and you have selected more than one drive from any one tray, the storage array returns an error. If you set `trayLossProtect` parameter to `FALSE`, the storage array performs operations, but the volume group that you create might not have tray loss protection.

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

[2] In trays that have drawers for holding the drives, drawer loss protection determines whether data on a volume is accessible or inaccessible if a drawer fails. To help make sure that your data is accessible, set the `drawerLossProtect` parameter to `TRUE`. For drawer loss protection to work, each drive in a volume group must be in separate drawers. If you have a storage array configuration in which a volume group spans several trays, you must make sure that the setting for drawer loss protection works with the setting for tray loss protection. If you set the `trayLossProtect` parameter to `TRUE`, you must set the `drawerLossProtect` parameter to `TRUE`. If you set the `trayLossProtect parameter` to `TRUE`, and you set the `drawerLossProtect parameter` to `FALSE`, the storage array returns an error message and a storage array configuration will not be created.

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

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

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

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

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

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

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

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

Naming Conventions

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

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

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

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

```
raidLevel=5
```

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

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

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

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

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

`numberOfDrives`

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

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

```
create volume drives=(0,2 0,3 1,4 1,5 2,6 2,7)
raidLevel=5
userLabel=”v1” capacity=2gb owner=a;
create volume volumeGroup=2 userLabel=”v2”
capacity=1gb owner=b;
create volume volumeGroup=2 userLabel=”v3”
capacity=1gb owner=a;

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

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.
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.

Status Value	Meaning
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;
```

Script Commands 2

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 script command 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" lists the general rules for entering the names of storage array entities, such as volumes or drives, with the script commands.
- "Formatting CLI Commands" lists the general formatting rules that apply to the CLI command wrapper.
- "Formatting Rules for Script Commands" lists the general formatting rules that apply to the script command syntax.
- "Command Reference – Listed Alphabetically" lists the script commands alphabetically and, for each script command, includes script command name, syntax, and parameters.
- "Command Reference - Listed by Function" lists the script commands organized into groups related to the physical features, the logical features, and the operational features of the storage array.

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.

Command Reference – Listed Alphabetically

Activate Asynchronous Mirroring

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

Syntax

```
activate storageArray feature=asyncRemoteMirror
```

Activate Host Port

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

Syntax

```
activate hostPort "userLabel"
```

Activate iSCSI Initiator

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

Syntax

```
activate iscsiInitiator "iscsiID"
```

Activate Storage Array Firmware

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

Syntax

```
activate storageArray firmware
```

Activate Synchronous Mirroring

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

- User-defined drives
- User-defined volume group
- 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.

NOTE In previous versions of this command the feature identifier was `remoteMirror`. This feature identifier is no longer valid and is replaced by `syncMirror`.

Syntax (User-Defined Drives)

```
activate storageArray feature=syncMirror
repositoryRAIDLevel=(1 | 3 | 5 | 6)
repositoryDrives=(trayID1,drawerID1,slotID1
... trayIDn,drawerIDn,slotIDn)
repositoryVolumeGroupUserLabel=[volumeGroupName]
driveMediaType=(HDD | SSD | unknown | allMedia)
driveType=(fibre | SATA | SAS)
[trayLossProtect=(TRUE | FALSE)
drawerLossProtect=(TRUE | FALSE)
dataAssurance=(none | enabled)]
```

Syntax (User-Defined Volume Group)

```
activate storageArray feature=syncMirror
repositoryVolumeGroup=volumeGroupName
[freeCapacityArea=freeCapacityIndexNumber]
```

Syntax (User-Defined Number of Drives)

```
activate storageArray feature=syncMirror
repositoryRAIDLevel=(1 | 3 | 5 | 6)
repositoryDriveCount=numberOfDrives
repositoryVolumeGroupUserLabel=[volumeGroupName]
driveMediaType=(HDD | SSD | unknown | allMedia)
driveType=(fibre | SATA | SAS)]
[trayLossProtect=(TRUE | FALSE)
drawerLossProtect=(TRUE | FALSE)
dataAssurance=(none | enabled)]
```

Add Drives to SSD Cache

Use this command to increase the capacity of an existing SSD cache by adding additional solid state disks (SSDs).

Syntax

```
set ssdCache [ssdCacheName]
addDrives=(trayID1,drawerID1,slotID1
... trayIDn,drawerIDn,slotIDn)
```

Add Member to Consistency Group

This command adds a new base volume as a member to an existing consistency group. You can specify an existing existing repository volume for the new consistency group member, or create a new repository volume. When you create a new repository volume, you identify an existing volume group or an existing disk pool where you want the repository volume.

Syntax for Use With an Existing Repository Volume

```
set consistencyGroup ["consistencyGroupName"]
addCGMemberVolume="baseVolumeName"
repositoryVolume="repos_XXXX"
```

Syntax for Use When Creating a New Repository Volume in a Volume Group

```
set consistencyGroup ["consistencyGroupName"]
addCGMemberVolume="baseVolumeName"
repositoryVolume=("volumeGroupName"
capacity=capacityValue(KB|MB|GB|TB|bytes))
```

Syntax for Use When Creating a New Repository Volume in a Disk Pool

```
set consistencyGroup ["consistencyGroupName"]
addCGMemberVolume="baseVolumeName"
repositoryVolume=("diskPoolName"
capacity=capacityValue(KB|MB|GB|TB|bytes))
```

Add Volume to Asynchronous Mirror Group

This command adds a primary volume to an asynchronous mirror group. This command is valid only on the local storage array that contains the asynchronous mirror group to which you want to add the primary volume. An asynchronous mirror group has a repository volume that is used to save data for all of the point-in-time images that are part of the asynchronous mirror group. Each primary volume in the asynchronous mirror group has a corresponding mirror volume on a remote storage array.

Syntax

```
add volume="volumeName"
asyncMirrorGroup="asyncMirrorGroupName"
remotePassword="password"
(repositoryVolume="repos_xxxx" |
repositoryVolume=(volumeGroupName [capacity=capacityValue])
repositoryVolume=(diskPoolName [capacity=capacityValue]))
```

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)
driveMediaType=(hdd | ssd | allMedia | unknown) |
raidLevel=(0 | 1 | 3 | 5 | 6)
volumeGroupWidth=numberOfDrives
volumeGroupCount=numberOfVolumeGroups
volumesPerGroupCount=numberOfVolumesPerGroup
hotSpareCount=numberOfHotSpares
segmentSize=segmentSizeValue
cacheReadPrefetch=(TRUE | FALSE)
readAheadMultiplier=multiplierValue
securityType=(none | capable | enabled)
dataAssurance=(none | enabled)]
```

Autoconfigure Storage Array Hot Spares

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

Syntax

```
autoConfigure storageArray hotSpares
```

Cancel Asynchronous Mirror Group Role Reversal

This command cancels a pending role reversal operation between asynchronous mirror groups.

Syntax

```
stop asyncMirrorGroup ["asyncMirrorGroupName"] rolechange
```

Change SSD Cache Application Type

This command changes the application type associated with the SSD cache. The application type can be web server, database, or multimedia. Changing the application type changes the block size, subblock size, populate on read threshold, and populate on write threshold for the volumes underlying the SSD cache.

Syntax

```
set ssdCache [ssdCacheName]
usageHint=(webServer|dataBase|fileSystem)
```

Check Asynchronous Mirror Group Consistency

The command produces a report based on analysis of the data in the repository. This command applies to an asynchronous mirror group that has underlying repository volumes.

Syntax

```
check asyncMirrorGroup[asyncMirrorGroupName]
repositoryConsistency localVolume=["localVolumeName"]
file="filePath"
```

Check Repository Consistency

This command applies to a number of objects that have underlying repository volumes. The command produces a report based on analysis of the data in the repository.

Syntax

```
check [snapGroup[snapGroupName] repositoryConsistency |
snapVolume[snapVolumeName] repositoryConsistency |
volume[volumeName] repositoryConsistency |
volumeCopy target[targetName] repositoryConsistency |
asyncMirrorGroup[asyncMirrorGroupName] repositoryConsistency
localVolume="localVolumeName"]
file="filePath"
```

Check Storage Array Connectivity

This command verifies that the local storage array and the remote storage array have a communication path and displays the connection details between the local and remote storage array.

Before creating an asynchronous mirror group, you should check whether the local storage array and the remote storage array can communicate with each other. When you execute this command, the system queries for the remote storage array on all eligible host ports to determine what ports are connected to the remote storage array. The result of the test is a list of all ports on the local storage array along with a list of the remote storage array port addresses accessible through that port.

NOTE Connectivity is tested using all possible channels, and if it is a dual controller configuration, connectivity is checked from each controller. It might take up to 20 minutes to check connectivity between two storage arrays.

Syntax

```
check storageArray connectivity
(remoteStorageArrayName="storageArrayName" |
remoteStorageArrayWwid=<wwID>)
```

Check Synchronous Mirroring 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.

NOTE In previous versions of this command the feature identifier was `remoteMirror`. This feature identifier is no longer valid and is replaced by `syncMirror`.

Syntax

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

Check Volume Parity

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

Syntax

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

Clear Asynchronous Mirroring Fault

This command clears an asynchronous mirroring "sticky" fault from one or more asynchronous mirror groups and one or more asynchronous mirror group member volumes.

An asynchronous mirror group and its member volumes can encounter asynchronous mirroring "sticky" faults, which occur at a single point-in-time but do not impact the functionality of the mirrors. These type of faults must be reviewed, but might or can? not require any changes to the configuration.

An asynchronous mirror group and its member volumes might or can have more than one associated sticky fault. This command clears all of the faults associated with the asynchronous mirror group and its member volume. However, if an asynchronous mirror group has a fault and one of its member volumes has a fault, clearing the fault on the asynchronous mirror group does not clear the fault on its member volume.

Syntax

```
clear asyncMirrorFault(all | asyncMirrorGroup
["asyncMirrorGroupName"] |
asyncMirrorGroups ["asyncMirrorGroupName1"
... "asyncMirrorGroupNameN"] |
volume ["volumeName"] |
volumes ["volumeName1" ... "volumeNameN"])
```

Clear Drive Channel Statistics

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

Syntax

```
clear all DriveChannels stats
```

Clear Storage Array Configuration

Use this command to perform one of these operations:

- Clear the entire storage array configuration, and return it back to the initial installation state
- Clear the configuration except for security information and identification information
- Clear volume group 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 | volumeGroups]
```

Clear Storage Array Core Dump

This command sets a flag on a controller to allow a new core dump to overwrite an existing core dump.

Syntax

```
set storageArray coreDumpAllowOverWrite
```

Clear Storage Array Event Log

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

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

Syntax

```
clear storageArray eventLog
```

Clear Storage Array Firmware Pending Area

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

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

Syntax

```
clear storageArray firmwarePendingArea
```

Clear Storage Array Recovery Mode

This command forces a storage array to exit recovery mode.

Syntax

```
clear storageArray recoveryMode
```

Clear Volume Reservations

This command clears persistent volume reservations.

Syntax

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

Clear Volume Unreadable Sectors

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

Syntax

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

Configure Automatic Support Bundle Collection

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

This command enables or disables the automatic collection of support bundles on one or more storage arrays when a critical MEL event is detected.

Syntax

```
SMcli -supportBundle auto (enable|disable)
(all | storageArrayName)
[data=fileName]
```

Convert Snapshot (Legacy) Volume to Snapshot Group

This command migrates from a snapshot (legacy) volume to a snapshot group, and converts from an existing snapshot (legacy) repository volume to a snapshot image repository volume. A maximum of four snapshots (legacy) can be converted to a snapshot group. The snapshots (legacy) must be in a Disabled state and must not be involved in a volume copy operation. The base volume can have only four snapshots (legacy). If the base volume has more than four snapshots (legacy), you must delete the extra snapshots (legacy) before running this command.

NOTE You must disable any snapshots (legacy) before attempting to convert from a snapshot (legacy) volume to a snapshot group. Use the `disableSnapshot` command to disable a snapshot (legacy).

Syntax

```
convert snapshotVolume baseVolume="baseVolumeName"
```

Create Asynchronous Mirror Group

This command creates a new, empty asynchronous mirror group on both the local storage array and the remote storage array. An asynchronous mirror group is a container that can house several mirrored pairs so that they can be managed as one entity. You create an asynchronous mirror group to define the synchronization settings for all mirrored pairs within the group. Each mirrored pair in an asynchronous mirror group share the same synchronization settings, primary and secondary role, and write mode.

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

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

Syntax

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

Create Consistency Group

NOTE This command does not apply to the snapshot (legacy) commands.

This command creates a new, empty consistency group that can contain snapshot groups. You must add the snapshot groups using the `set consistencyGroup addCGMember` command.

Syntax

```
create consistencyGroup userLabel="consistencyGroupName"
[repositoryFullPolicy=(failBaseWrites | purgeSnapImages) |
repositoryFullLimit=percentValue |
autoDeleteLimit=numberOfSnapImages |
enableSchedule=(TRUE | FALSE) |
schedule (immediate | snapSchedule) |
rollbackPriority=(lowest | low | medium | high | highest)]
```

Create Consistency Group Snapshot Image

This command creates a new snapshot image for each base volume that is a member of a snapshot consistency group.

Syntax

```
create cgSnapImage consistencyGroup="consistencyGroupName"
```

Create Consistency Group Snapshot Volume

This command creates a snapshot volume of specific images in the base volumes in a consistency group. You can select one base volume or more than one base volumes from the consistency group to include in the snapshot volume. When you create a snapshot volume of a consistency group you are creating a volume with contents that you can view.

Syntax With User Specified Base Volumes

```
create cgSnapVolume userLabel="cgVolumeName"
cgSnapImageID="snapCGID:imageID"
members=(baseVolume1:repos_XXXX ... baseVolumen:repos_YYYY)
```

Syntax When Setting the Consistency Group Snapshot Volume to Read Only

```
create cgSnapVolume userLabel="cgVolumeName"
cgSnapImageID="snapCGID:imageID"
readOnly
```

Syntax When Setting the Repository Full Limit

```
create cgSnapVolume userLabel="cgVolumeName"
cgSnapImageID="snapCGID:imageID"
members=(baseVolume1:repos_XXXX | baseVolume1:
(volumeGroupName [capacity=capacityValue])|(
baseVolume1:diskPoolName
[capacity=capacityValue]) ... baseVolumen:repos_YYYY |
baseVolumen:
(volumeGroupName [capacity=capacityValue])| baseVolumen:
(diskPoolName
[capacity=capacityValue])) repositoryFullLimit=percentValue
```

Create Consistency Group Snapshot Volume Mapping

This command creates a logical mapping from a consistency group snapshot volume to a host or a host group.

Syntax

```
create mapping cgSnapVolume="snapVolumeName"
(host="hostName" | hostGroup=("hostGroupName" |
defaultGroup)
```

Create Disk Pool

This command creates a new disk pool based on the specified parameters. You can create the disk pool by entering either a list of drives or a type of drive that you want to use for the disk pool.

NOTE If you enter a list of drives, make sure that all of the drives have the same capacity. If the drives do not have the same capacity each drive in the disk pool reports capacity equal to the smallest drive.

Syntax

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

Create Host

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

Syntax

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

Create Host Group

This command creates a new host group.

Syntax

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

Create Host Port

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

Syntax

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

Create iSCSI Initiator

This command creates a new iSCSI initiator object.

Syntax

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

Create RAID Volume (Automatic Drive Select)

This command creates a volume group across the drives in the storage array and a new volume in the volume group. 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
volumeGroupUserLabel="volumeGroupName"
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)
trayLossProtect=(TRUE | FALSE)
drawerLossProtect=(TRUE | FALSE)
dssPreAllocate=(TRUE | FALSE)
securityType=(none | capable | enabled)
dataAssurance=(none | enabled)]
```

Create RAID Volume (Free Extent Based Select)

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

Syntax

```
create volume volumeGroup="volumeGroupName"
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)
dataAssurance=(none | enabled)]
```

Create RAID Volume (Manual Drive Select)

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

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

Syntax

```
create volume drives=(trayID1,drawerID1,slotID1
... trayIDn,drawerIDn,slotIDn)
volumeGroupUserLabel="volumeGroupName"
raidLevel=(0 | 1 | 3 | 5 | 6)
userLabel="volumeName"
[capacity=volumeCapacity
owner=(a | b)
cacheReadPrefetch=(TRUE | FALSE)
segmentSize=segmentSizeValue
usageHint=(fileSystem | dataBase | multiMedia)
trayLossProtect=(TRUE | FALSE)
drawerLossProtect=(TRUE | FALSE)
dssPreAllocate=(TRUE | FALSE)
securityType=(none | capable | enabled)
dataAssurance=(none | enabled)]
```

Create Read-Only Snapshot Volume

This command creates a read-only snapshot volume for the snapshot images of a base volume. To change a read-only snapshot volume to a read/write volume, use the `set snapVolume convertToReadWrite` command.

NOTE You cannot use this command for a snapshot image that is used in online volume copy.

Syntax

```
create snapVolume userLabel="snapVolumeName"
snapImageID="snapCGID:imageID"
readOnly
```

Create Snapshot (Legacy) Volume

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

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

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

Syntax (User-Defined Drives)

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

Syntax (User-Defined Number of Drives)

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

Syntax (Existing Repository Volume Group)

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

Create Snapshot Group

This command creates a new snapshot group and the associated repository volume. A snapshot group contains a sequence of snapshot images of an associated base volume. A snapshot group has a repository volume that is used to save data for all of the snapshot images that are part of the snapshot group.

Syntax

```
create snapGroup userLabel="snapGroupName"
sourceVolume="volumeName"
(repositoryVolume="repos_xxxx" |
repositoryVolume=(volumeGroupName [capacity=capacityValue])
repositoryVolume=(diskPoolName [capacity=capacityValue]))
[repositoryFullPolicy=(failBaseWrites | purgeSnapImages) |
rollbackPriority=(highest | high | medium | low | lowest) |
repositoryFullLimit=percentValue |
autoDeleteLimit=numberOfSnapImages |
enableSchedule=(TRUE | FALSE)
schedule (immediate | snapshotSchedule)]
```

Create Snapshot Image

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

Syntax

```
create snapImage (snapGroup="snapGroupName" |
snapGroups=("snapGroupName1"... "snapGroupNamen"))
```

Create Snapshot Volume

This command creates a snapshot volume with read-write capabilities for snapshot images of a base volume. You can map the snapshot volume to a host and all of the host writes reside in the repository volume associated with the snapshot volume. You can assign the new snapshot volume to an existing repository volume, or you can create a new repository volume in a volume group or disk pool.

NOTE You cannot use this command for a snapshot image that is used in online volume copy.

Syntax

```
create snapVolume userLabel="snapVolumeName"
snapImageID="snapCGID:imageID"
[(repositoryVolume="repos_xxxx" |
repositoryVolume=(volumeGroupName [capacity=capacityValue])
repositoryVolume=(diskPoolName [capacity=capacityValue]))
repositoryFullLimit=percentValue]
```

Create SSD Cache

This command creates a read cache for a storage array using Solid State Disks (SSDs). Using high performance SSDs to cache read data improves the application I/O performance and response times, and delivers sustained performance improvement across different workloads, especially for high-IOP workloads. SSD cache maximizes the use of expensive fast SSDs.

SSD cache works in addition to the primary cache in the controller DRAM. With controller cache, the data is stored in DRAM after a host read. With SSD cache, the data is copied from user-specified base volumes, and then cached on SSDs.

Syntax

```
create ssdCache userLabel="ssdCacheName"
drives=(trayID1,drawerID1,slotID1 ...
trayIDn,drawerIDn,slotIDn)
[updateExistingVolumes=(TRUE|FALSE)]
```

Create Storage Array Security Key

This command creates a new security key for a storage array that has full disk encryption (FDE) drives. This command also sets the security definitions and sets the state to Security Enabled. To use this command successfully, you need to have enough FDE drives to create at least one volume group or one disk pool.

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

Syntax

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

Create Synchronous Mirroring

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

NOTE In previous versions of this command the feature identifier was `remoteMirror`. This feature identifier is no longer valid and is replaced by `syncMirror`.

Syntax

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

Create Volume in Disk Pool

This command creates a new standard RAID volume or a thin volume in an existing disk pool.

NOTE Some parameters for creating a standard RAID volume in a volume group are not compatible for creating volumes of any type in a disk pool. When using older scripts to create volumes in disk pools, make sure that all of the parameters are valid for disk pools. Invalid parameters prevent the scripts from running correctly and cause an error to be posted.

Syntax for Creating a Standard Volume

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

Syntax for Creating a Thin Provisioned Volume

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

Create Volume Copy

This command creates a volume copy and starts the volume copy operation. This command is valid for both snapshot (legacy) volume copy pairs and snapshot image volume copy pairs.

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 (legacy) volumes or snapshot image 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 (legacy) also called *offline* volume copy
- Volume copy with either snapshot (legacy) or snapshot image, also called *online* volume copy

If you use volume copy without either snapshot (legacy) or snapshot image, 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 (legacy)or snapshot image. You can select volume copy with snapshot (legacy) or snapshot image through the optional parameters in the command syntax.

After completion of the volume copy with snapshot (legacy) operation, the snapshot (legacy) is disabled. After completion of the volume copy with snapshot image operation, the snapshot image is deleted and the snapshot volume 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)]
```

Create Volume Group

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

Syntax

```
create volumeGroup
drives=(trayID1,drawerID1,slotID1
... trayIDn,drawerIDn,slotIDn)
raidLevel=(0 | 1 | 3 | 5 | 6)
userLabel="volumeGroupName"
[driveMediaType=(HDD | SSD | unknown | allMedia)
driveType=(fibre | SATA | SAS)
trayLossProtect=(TRUE | FALSE)
drawerLossProtect=(TRUE | FALSE)
securityType=(none | capable | enabled)
dataAssurance=(none | enabled)]
```

Deactivate Asynchronous Mirroring

This command deactivates the Asynchronous Mirroring premium feature.

NOTE All existing asynchronous mirror groups or asynchronous mirrored pairs must be deleted from the local storage array and the remote storage array before the Asynchronous Mirroring feature can be deactivated.

Syntax

```
deactivate storageArray feature=asyncRemoteMirror
```

Deactivate Synchronous Mirroring

This command deactivates the Synchronous Mirroring 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.

NOTE In previous versions of this command the feature identifier was `remoteMirror`. This feature identifier is no longer valid and is replaced by `syncMirror`.

Syntax

```
deactivate storageArray feature=syncMirror
```

Delete Asynchronous Mirror Group

This command deletes one or more asynchronous mirror groups from the local storage array and the remote storage array.

NOTE The asynchronous mirror group must be empty before it can be successfully deleted. You must remove all asynchronous mirrored pairs from the asynchronous mirror group before using this command.

Syntax

```
delete asyncMirrorGroup
(allAsyncMirrorGroups |
asyncMirrorGroup["asyncMirrorGroupName"] |
asyncMirrorGroups
["asyncMirrorGroupName_01""asyncMirrorGroupName_02"])
```

Delete Consistency Group

This command deletes a snapshot consistency group. This command works in two ways:

- You can delete both the consistency group and the repository volumes contained by the consistency group.
- You can delete only the consistency group and leave the repository volumes that are contained by the consistency group intact.

Syntax

```
delete consistencyGroup [consistencyGroupName
[deleteRepositoryMembers=(TRUE | FALSE)]]
```

Delete Consistency Group Snapshot Image

This command deletes the snapshot images in a consistency group.

Syntax

```
delete cgSnapImage consistencyGroup="consistencyGroupName"
[(deleteCount=numberOfSnapImages |
retainCount=numberOfSnapImages) |
ignoreSnapVolume=(TRUE | FALSE)
```

Delete Consistency Group Snapshot Volume

This command deletes the snapshot volume of a consistency group. Optionally, you can also delete the repository members.

Syntax

```
delete cgSnapVolume ["snapVolumeName"]
[deleteRepositoryMembers=(TRUE | FALSE)]
```

Delete Disk Pool

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

This command deletes a disk pool. Depending on your version of the storage management software, this command also deletes all of the volumes in the disk pool. If your version of the storage management software does not support automatic deletion of the volumes, you can force the deletion of the disk pool and volumes.

Syntax

```
delete diskPool [diskPoolName]
[force=(TRUE | FALSE))]
```

Delete Host

This command deletes one or more hosts.

Syntax

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

Delete Host Group

This command deletes a host group.

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

Syntax

```
delete hostGroup [hostGroupName]
```

Delete Host Port

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

Syntax

```
delete hostPort [hostPortName]
```

Delete iSCSI Initiator

This command deletes a specific iSCSI initiator object.

Syntax

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

Delete Snapshot (Legacy) Volume

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

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 (volume [volumeName] |
volumes ["volumeName1"a ... "volumeNameN"])
[schedule]
```

Delete Snapshot Group

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

This command deletes an entire snapshot group and optionally the associated repository volumes.

Syntax

```
delete snapGroup ["snapGroupName"]
[deleteRepositoryMembers=(TRUE | FALSE)
```

Delete Snapshot Image

This command deletes one or more snapshot images from a snapshot group.

Syntax

```
delete snapImage (snapGroup="snapGroupName" |
snapGroups=("snapGroupName1"... "snapGroupNamen"))
(deleteCount=numberOfSnapImages |
retainCount=numberOfSnapImages |
ignoreSnapVolume=(TRUE | FALSE) |
snapImageID=OLDEST)
```

Delete Snapshot Volume

This command deletes a snapshot volume and optionally the associated snapshot repository members.

NOTE You cannot use this command for snapshot images involved in online volume copy.

Syntax

```
delete snapVolume ["snapVolumeName"]
[deleteRepositoryMembers=(TRUE | FALSE)]
```

Delete SSD Cache

This command deletes the SSD cache. All data in the SSD cache is purged.

Syntax

```
delete ssdCache [ssdCacheName]
```

Delete Volume

This command deletes one or more standard volumes, snapshot (legacy) volumes, or snapshot (legacy) 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])
[removeVolumeGroup=(TRUE | FALSE)
force=(TRUE | FALSE)]
```

Delete Volume from Disk Pool

This command deletes either normal or thin volumes from a disk pool. You also can use this command to delete any schedules related to the volume. When you delete the schedule the volume is not deleted.

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] | allVolumes)
[removeVolumeGroup=(TRUE | FALSE)
force=(TRUE | FALSE)
schedule
retainRepositoryMembers=(TRUE | FALSE)]
```

Delete Volume Group

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

This command deletes an entire volume group and its associated volumes.

Syntax

```
delete volumeGroup [volumeGroupName]
```

Diagnose Controller

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

Syntax

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

Diagnose Synchronous Mirroring

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

NOTE In previous versions of this command the feature identifier was `remoteMirror`. This feature identifier is no longer valid and is replaced by `syncMirror`.

Syntax

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

Disable AutoSupport at the EMW Level SMcli Version

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

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

Syntax

```
SMcli disable autoSupportFeature
```

Disable External Security Key Management

This command disables external security key management for a storage array that has full disk encryption drives.

Syntax

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

Disable Snapshot (Legacy)

This command stops a copy-on-write operation. This command performs the same action as the legacy `stop snapshot` command.

Syntax

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

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=featureAttributeList)
```

The *`feaatureAttributeList`* can be one or more of these attribute values. If you enter more than one attribute value, separate the values with a white space.

- `volumeCopy`
- `snapshot`
- `asyncMirror`
- `syncMirror`
- `mixedDriveTypes`
- `goldKey`
- `driveSecurity`
- `enterpriseSecurityKeyMgr`
- `thinProvisioning`
- `storagePartition[2|4|8|16|32|64|96|128|192|256|512|Max]`

  The form for this attribute value is a combination of the alphabetical term merged with the numerical value, as shown by this example: `storagePartition256`
- `driveSlotLimit[16|24|32|48|60|64|72|96|112|10|128|136|144|180|192|256| 272|300|360|384|448|480|Max]`

  The form for this attribute value is a combination of the alphabetical term merged with the numerical value, as shown by this example: `driveSlotLimit360`
- `ssdCache`
- `dataAssurance`
- `SSDSupport`
- `highPerformanceTier`
- `raid6`

NOTE The folllowing attribute is for enabling the snapshot (legacy) feature only.

- `snapshot[2|4|8|16]`

  The form for this attribute value is a combination of the alphabetical term merged with the numerical value, as shown by this example: `snapshot16`

NOTE The `remoteMirror` attribute is valid only for firmware versions before 7.84. Starting with firmware 7.84 the `remoteMirror` attribute is replaced by `syncMirror`.

- `remoteMirror[8|16|32|64|128]`

  The form for this attribute value is a combination of the alphabetical term merged with the numerical value, as shown by this example: `remoteMirror128`

Display Automatic Support Bundle Collection Configuration

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

---

This command displays the automatic support bundle collection settings.

Syntax

```
SMcli -supportBundle auto show
```

Display Automatic Support Bundle Collection Schedule

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

---

This command displays the schedule for collecting support bundles for all storage arrays.

Syntax

```
SMcli -supportBundle schedule show
```

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 [trayID,drawerID,slotID] firmware
file="filename"
```

Download Environmental Card Firmware

This command downloads environmental services module (ESM) firmware.

Syntax

```
download (allTrays | tray [trayID])
firmware file="filename"
```

Download Storage Array Drive Firmware

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

Syntax

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

Download Storage Array Firmware/NVSRAM

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

Syntax

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

Download Storage Array NVSRAM

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

Syntax

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

Download Tray Configuration Settings

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

Syntax

```
download (allTrays | tray [trayID]) configurationSettings
file="filename"
```

Enable AutoSupport at the EMW Level SMcli Version

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

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

Syntax

```
SMcli enable autoSupportFeature
```

Enable Controller Data Transfer

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

Syntax

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

Enable Disk Pool Security

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

NOTE All of the drives that comprise the disk pool must be security capable.

Syntax

```
enable diskPool [diskPoolName] security
```

Enable External Security Key Management

This command enables external security key management for a storage array that has full disk encryption drives.

Syntax

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

Enable or Disable SSD Cache for a Volume

This command turns on or off caching using the SSD cache feature for a specific volume. The volume can be either a standard volume, a snapshot volume, or a consistency group snapshot volume.

Syntax Applicable to a Standard Volume

```
set volume ["volumeName"] ssdCacheEnabled=(TRUE | FALSE)
```

Syntax Applicable to a Snapshot Volume

```
set snapVolume ["snapVolumeName"] ssdCacheEnabled=(TRUE |
FALSE)
```

Syntax Applicable to a Consistency Group Snapshot Volume

```
set cgSnapVolume ["cgSnapVolumeName"] ssdCacheEnabled=(TRUE
| FALSE)
```

Enable Storage Array Feature

This command enables a premium feature for either a permanent upgrade to the storage array or a trial period. This command performs one of these actions:

- Enables a feature key for a permanent upgrade of a feature
- Enables a feature key for a permanent upgrade of a feature pack
- Enables a feature for a trial period

A feature pack is a predefined set of several premium features, such as Storage Partitioning and Synchronous Mirroring. These premium features are combined for the convenience of the users. When users install a feature pack, all of the premium features in the feature pack are installed at one time.

Each premium feature is managed by a license key that is generated for a specific feature or feature pack and a specific storage array. The license key is delivered as a file that you run to apply the license for the feature.

To determine which features are loaded on the storage array run the `show storageArray features` command. The `show storageArray features` command lists all of the premium features installed on the storage array, which premium features can be evaluated for a trial period, which premium features are enabled, and which premium features are disabled.

Syntax to Enable a Feature Key

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

The `file` parameter identifies 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.

You will need a feature key file for each premium feature that you want to enable.

Syntax to Enable a Feature Pack

```
enable storageArray featurePack file="filename"
```

The `file` parameter identifies the file path and the file name of a valid feature pack file. Enclose the file path and the file name in double quotation marks (" "). For example:

```
file="C:\Program Files\CLI\dnld\ftrpk.key"
```

Valid file names for feature key files end with a `.key` extension.

Syntax to Enable a Feature for a Trial Period

```
enable storageArray feature=featureAttributeList
```

To evaluate a feature for a trial period, you can enter one or more of the following attribute values for the *`featureAttributeList`*. If you enter more than one attribute value, separate the values with a white space.

- `asyncMirror`
- `syncMirror`
- `snapshot`
- `ssdCache`
- `volumeCopy`
- `thinProvisioning`

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.

- `highPerformanceTier`
- `SSDSupport`

Enable Volume Group Security

This command converts a non-secure volume group to a secure volume group.

Syntax

```
enable volumeGroup [volumeGroupName] security
```

Establish Asynchronous Mirrored Pair

Use this command to complete an asynchronous mirrored pair on the remote storage array by adding a secondary volume to an existing asynchronous mirror group. Beforeyou run this command, the asynchronous mirror group must exist and the primary volume must exist in the asynchronous mirror group. After this command successfully completes, asynchronous mirroring starts between the primary volume and the secondary volume.

The two volumes comprising an asynchronous mirrored pair function as a single entity. Establishing an asynchronous mirrored pair allows you to perform actions on the entire mirrored pair versus the two individual volumes.

Syntax

```
establish asyncMirror volume="secondaryVolumeName"
asyncMirrorGroup="asyncMirrorGroupName"
primaryVolume="primayVolumeName"
```

Export Storage Array Security Key

This command saves a full disk encryption (FDE) security key to a file. You can transfer the file from one storage array to another storage array. The file enables you to move FDE drives between storage arrays.

Syntax

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

Import Storage Array Security Key

This command unlocks one or more full disk encryption (FDE) drives that you have imported from one storage array to another storage array. Only the FDE 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"
```

Increase Capacity of Volume in Disk Pool

This command increases the capacity of either a standard volume or a repository volume in a disk pool. In this command, a standard volume is also called a thick volume.

NOTE You cannot use this command to increase the capacity of a thin volume.

Syntax

```
start increaseVolumeCapacity volume="volumeName"
incrementalCapacity=volumeCapacity
[addDrives=(trayID1,drawerID1,slotID1 ...
trayIDn,drawerIDn,slotIDn)]
```

Initialize Thin Volume

This command initializes or re-initializes a thin volume.

- Used without any of the optional parameters, this command deletes the data on the thin volume. The repository volume capacity is not affected.
- Used with any of the optional parameters, this command cause reinitialization and repository volume actions.

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

Syntax

```
start volume [volumeName] initialize
[existingRepositoryLabel=existingRepositoryName |
diskPool="diskPoolName" capacity=capacityValue|
retainRepositoryMembers=[TRUE|FALSE]]
```

Load Storage Array DBM Database

This command restores a Database Management (DBM) database image by retrieving the image from a file or from cache. This command restores a storage array DBM database to the exact configuration that existed when the database image was captured using the `save storageArray dbmDatabase` command. The data in a file can be just RAID configuration data or all data, including data for RAID configuration, volume groups and disk pools. The data in a cache location always includes all data.

Before using this command with the `file` option, you must first obtain a validator string (a security code) from your Technical Support Representative. To obtain a validator, use the `save storageArray dbmValidator` command to generate an XML file that contains validator information. Your Technical Support Representative uses the XML file to generate the validator string required for this command.

Syntax

```
load storageArray dbmDatabase
((file="filename" validator="validatorValue") |
sourceLocation=onboard)
[controller [(a|b)] |
contentType=(partial | all)]
```

Locate SSD Cache

The `start locate` command identifies the Solid State Disks (SSDs) that are being used in the SSD cache by turning on the indicator lights for the SSDs. The `stop locate` command turns off the indicator lights on the SSDs.

Syntax Applicable to Starting a Locate Operation

```
start ssdCache [ssdCacheName] locate
```

Syntax Applicable to Stopping a Locate Operation

```
stop ssdCache locate
```

Recopy Volume Copy

NOTE With firmware version 7.83 the `copyType=(online | offline)` parameter is no longer used.

This command reinitiates a volume copy operation using an existing volume copy pair. This command is valid for both snapshot (legacy) volume copy pairs and snapshot image volume copy pairs.

This command works with volume copy pairs that you created with a snapshot (legacy) volume or with a snapshot image volume.

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 (legacy) volumes or snapshot image 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.

Syntax

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

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 volume group or create a new volume group 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=(trayID,drawerID,slotID) |
drives=(trayID1,drawerID1,slotID1
... trayIDn,drawerIDn,slotIDn) |
volumeGroup=volumeGroupName))
[newVolumeGroup=volumeGroupName]
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)
dataAssurance=(none | enabled)]
```

Re-create External Security Key

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

Syntax

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

Re-create Snapshot (Legacy)

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

Syntax

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

Re-create Synchronous Mirroring 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 Synchronous Mirroring 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 volume group, 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=(trayID1,slotID1 ... trayIDn,slotIDn)
[trayLossProtect=(TRUE | FALSE)
dataAssurance=(none | enabled)]
```

Syntax (User-Defined Volume Group)

```
recreate storageArray mirrorRepository
repositoryVolumeGroup=volumeGroupName
[freeCapacityArea=freeCapacityIndexNumber]
```

Syntax (User-Defined Number of Drives)

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

Remove Drives from SSD Cache

This command decreases the capacity of the SSD cache by removing Solid State Disks (SSDs).

Syntax

```
set ssdCache [ssdCacheName]
removeDrives=(trayID1,drawerID1,slotID1 ...
trayIDn,drawerIDn,slotIDn)
```

Remove Incomplete Asynchronous Mirrored Pair from Asynchronous Mirror Group

This command removes an orphaned mirrored pair volume on the storage array. An orphaned mirrored pair volume exists when a member volume in an asynchronous mirror group has been removed on one side of the asynchronous mirror group (either the primary side or secondary side) but not on the other side.

Orphan mirrored pair volumes are detected when inter-controller communication is restored and the two sides of the mirror configuration reconcile mirror parameters.

Use this command when the mirror relationship is successfully removed on the local or remote storage array, but cannot be removed on the corresponding storage array because of a communication problem.

Syntax

```
remove asyncMirrorGroup ["asyncMirrorGroupName"]
incompleteMirror volume="volumeName"
```

Remove Member Volume from Consistency Group

This command removes a member volume from a an existing snapshot consistency group. Optionally, you can delete the repository volume members from the consistency group.

Syntax

```
set consistencyGroup ["consistencyGroupName"]
removeCGMemberVolume="memberVolumeName"
[deleteRepositoryMembers=(TRUE | FALSE)]
```

Remove Synchronous Mirroring

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

NOTE In previous versions of this command the feature identifier was `remoteMirror`. This feature identifier is no longer valid and is replaced by `syncMirror`.

Syntax

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

Remove Volume Copy

NOTE With firmware version 7.83 the `copyType=(online | offline)` parameter is no longer used.

This command removes a volume copy pair. This command is valid for both snapshot (legacy) volume copy pairs and new snapshot volume copy pairs.

Syntax

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

Remove Volume from Asynchronous Mirror Group

This command removes a member volume from an existing asynchronous mirror group. Optionally, you can delete the repository volume members from the asynchronous mirror group.

This command is valid only on the local storage array that contains the asynchronous mirror group whose member volume that you want to remove.

Syntax

```
remove volume ["volumeName"]
asyncMirrorGroup="asyncMirrorGroupName"
[deleteRepositoryMembers=(TRUE | FALSE)]
```

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))
```

Rename Snapshot Volume

This command renames an existing snapshot volume.

Syntax

```
set snapVolume ["snapVolumeName"]
userLabel="snapImageVolumeName"
```

Rename SSD Cache

This command changes the name of the SSD cache.

Syntax

```
set ssdCache [old_ssdCacheName] userLabel="new_ssdCacheName"
```

Repair Volume Parity

This command repairs the parity errors on a volume.

Syntax

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

Replace Drive

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

Syntax

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

Reset Asynchronous Mirror Group Statistics

This command resets the synchronization statistics for one or more member volumes in an asynchronous mirror group to a relative 0.

Syntax

```
reset storageArray arvmStats asyncMirrorGroup
["asyncMirrorGroupName"]
volume="volumeName" sampleType=(all | mostRecent |
longestSyncTime | errors)
```

Reset Controller

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

ATTENTION When you reset a controller, the controller is removed from the data path and is not available for I/O operations until the reset operation is complete. If a host is using volumes that are owned by the controller being reset, the I/O directed to the controller is rejected. Before resetting the controller, either make sure that the volumes that are owned by the controller are not in use or make sure that a multi-path driver is installed on all of the hosts that use these volumes.

Syntax

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

Notes

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

Reset iSCSI IP Address

This command resets the IP address for the remote storage array to re-establish connection with the local storage array. You can use this command to notify the local storage array that the iSCSI IP address of the remote storage array have changed and need to be updated.

When establishing an asynchronous mirroring relationship with an iSCSI connection, both storage arrays store a record of the IP address of the remote storage array of the asynchronous mirroring configuration. If the IP address of an iSCSI port changes, the remote storage array that is attempting to use that port encounters a communication error.

The storage array with the changed IP address sends a message to each remote storage array associated with the asynchronous mirror groups that are configured to mirror over an iSCSI connection. Storage arrays that receive this message automatically update their remote-target IP address.

If the storage array with the changed IP address is unable to send its inter-controller message to a remote storage array, the system sends you an alert of the connectivity issue. Use the `reset` command to re-establish connection with the local storage array.

Syntax

```
reset (remoteStorageArrayName="storageArrayName" |
remoteStorageArrayWwid=<wwID>)
iscsiIpAddress
```

Reset Storage Array Battery Install Date

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

Syntax

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

Reset Storage Array Diagnostic Data

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

ATTENTION Run this command only with the assistance of your Technical Support Representative.

Syntax

```
reset storageArray diagnosticData
```

Reset Storage Array Infiniband Statistics Baseline

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

Syntax

```
reset storageArray ibStatsBaseline
```

Reset Storage Array iSCSI Baseline

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

Syntax

```
reset storageArray iscsiStatsBaseline
```

Reset Storage Array RLS Baseline

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

Syntax

```
reset storageArray RLSBaseline
```

Reset Storage Array SAS PHY Baseline

This command resets the SAS physical layer (SAS PHY) baseline for all devices, 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 previous release of the `reset storageArray SASPHYBaseline` command cleared error counts for all devices except the drives. The `reset storageArray SASPHYBaseline` command now resets the SAS PHY baseline for the drives as well as the other devices. All errors are deleted from the `.csv` file.

Syntax

```
reset storageArray SASPHYBaseline
```

Reset Storage Array SOC Baseline

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

Syntax

```
reset storageArray SOCBaseline
```

Reset Storage Array Volume Distribution

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

Syntax

```
reset storageArray volumeDistribution
```

Resume Asynchronous Mirror Group

This command resumes data transfer between all mirrored pairs in an asynchronous mirror group. Data written to the primary volumes while the asynchronous mirror group was suspended is written to the secondary volumes immediately. Periodic synchronization resumes if an automatic synchronization interval has been set.

Syntax

```
resume asyncMirrorGroup ["asyncMirrorGroupName"]
```

Resume Consistency Group Snapshot Volume

This command restarts a copy-on-write operation for creating a consistency group snapshot volume that you stopped using the `stop cgSnapVolume` command.

Syntax

```
resume cgSnapVolume ["snapVolumeName"]
cgSnapImage="snapImageName"
```

Resume Snapshot (Legacy) 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 (legacy) volume reverts to a paused state, and the Needs Attention condition is displayed.

Syntax

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

Resume Snapshot Image 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 causes a Needs Attention condition for the storage array.

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

NOTE You cannot use this command for snapshot images involved in online volume copy.

Syntax

```
resume snapImage [snapImageName] rollback
```

Resume Snapshot Volume

This command resumes a snapshot volume operation that was stopped.

Syntax

```
resume snapVolume [snapVolumeName]
snapImage="snapCGID:imageID"
```

Resume SSD Cache

This command restarts the caching for all of the volumes using the SSD cache that was temporarily stopped with the `suspend ssdCache` command.

Syntax

```
resume ssdCache [ssdCacheName]
```

Resume Synchronous Mirroring

This command resumes a suspended Synchronous Mirroring operation.

NOTE In previous versions of this command the feature identifier was `remoteMirror`. This feature identifier is no longer valid and is replaced by `syncMirror`.

Syntax

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

Revive Drive

This command forces the specified drive to the Optimal state.

ATTENTION Possible loss of data access – Correct use of this command depends on the data configuration on all of the drives in the volume group. Never try to revive a drive unless you are supervised by your Technical Support Representative.

Syntax

```
revive drive [trayID,drawerID,slotID]
```

Revive Snapshot Group

This command forces the specified snapshot group to the Optimal state. If the snapshot group is not in a Failed state, the firmware displays an error message and does not run this command.

Syntax

```
revive snapGroup ["snapGroupName"]
```

Revive Snapshot Volume

This command forces a specified snapshot volume to the Optimal state. The snapshot volume can be one of these:

- A standalone snapshot volume
- A snapshot volume that is a member of a consistency group

If the snapshot volume is not in a Failed state, the firmware displays an error message and does not run this command.

NOTE You cannot use this command for a snapshot volume that is used in online volume copy.

Syntax

```
revive snapVolume ["snapVolumeName"]
```

Revive Volume Group

This command forces the specified volume group 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 volume group. Never try to revive a drive unless you are supervised by your Technical Support Representative.

Syntax

```
revive volumeGroup [volumeGroupName]
```

Save Asynchronous Mirror Group Statistics

This command saves to a file the synchronization statistics for one or more member volumes in an asynchronous mirror group. The statistics collected are available only for asynchronous mirror group member volumes in a primary role on the local storage array.

A set of data is collected during the synchronization process that can be used to evaluate how well the mirror configuration is functioning. The data is collected as a set of *samples*. A sample is created at the beginning of a synchronization process and updated regularly while the synchronization process proceeds.

A sample collects data until the synchronization process completes or until a disruption to the synchronization process occurs such as a volume ownership transfer or a `read-write` error. When a synchronization process disruption is resolved (for example, the volume is transferred to the alternate controller), a new sample is created and updated as the synchronization process continues.

Syntax

```
save storageArray arvmStats asyncMirrorGroup
["asyncMirrorGroupName"]
arvmStats file="filename"
[volume="volumeName" |
sampleType=(all | mostRecent | longestSyncTime | errors) |
recordLimit=(1-90)]
```

Save Controller NVSRAM

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

Syntax

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

Save Drive Channel Fault Isolation Diagnostic Status

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

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

Syntax

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

Save Drive Log

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

Syntax

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

Save Storage Array Configuration

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

Syntax

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

Save Storage Array Core Dump

This command saves a core dump to a specified file on a host.

Syntax

```
save storageArray coreDump file="filename"
```

Save Storage Array DBM Database

Use this command to back up RAID configration data or all data to a file on the host. You can specify multiple data locations and controller.

Syntax

```
save storageArray dbmDatabase
file="fileName"
[sourceLocation=(disk | onboard) |
controller [(a|b)] |
contentType=(partial | all)]
```

Save Storage Array DBM Validator Information File

This command saves the database management (DBM) validation information for a storage array in an XML file. A Technical Support Representative can use this XML file to generate a validator string (a security code). The validator string must be included in the `load storageArray dbmDatabase` command when restoring a storage array back to a pre-existing configuration.

Syntax

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

Save Storage Array Diagnostic Data

This command saves the storage array diagnostic data from either the controllers or the environmental services modules (ESMs) to a file. You can review the file contents at a later time. You can also send the file to your Technical Support 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 Technical Support Representative.

Syntax

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

Save Storage Array Events

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

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

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

Syntax

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

Save Storage Array Firmware Inventory

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

- Controllers
- Drives
- Drawers (if applicable)
- Environmental services modules (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 Technical Support Representative for further review.

Syntax

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

Save Storage Array InfiniBand Statistics

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

Syntax

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

Save Storage Array iSCSI Statistics

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

Syntax

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

Save Storage Array Performance Statistics

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

Syntax

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

Save Storage Array RLS Counts

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

Syntax

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

Save Storage Array SAS PHY Counts

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

Syntax

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

Save Storage Array SOC Counts

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

Syntax

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

Save Storage Array State Capture

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

Syntax

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

Save Storage Array Support Data

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

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

Syntax

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

Save Tray Log

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

Syntax

```
save allTrays logFile="filename"
```

Schedule Automatic Support Bundle Collection Configuration

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

This command creates or removes a schedule for saving a support bundle on one or more storage arrays.

Syntax

```
SMcli -supportBundle schedule (enable|disable)
(all|storageArrayName)
[data=pathName |
startTime=HH:MM |
startDate=MM:DD:YYYY] |
endDate=MM:DD:YYYY] |
(daysOfWeek=(Sunday Monday Tuesday Wednesday Thursday Friday
Saturday) |
dayOfWeek=(Sunday|Monday|Tuesday|Wednesday|Thursday|Friday|S
aturday)|
months=(January February March April May June July August
September October November December)
onDays=(1-31) |
weekNumber=(First|Second|Third|Fourth|Last))
```

Set Alert Severities

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

This command enables you to set the level of severities that casuses an alert to be sent to the Windows event log. The alert severities apply to all of the storage arrays in the entire storage system.

Syntax

```
SMcli -alertSeverities (severity |
[severity1, ... severityN])
```

Set Asynchronous Mirror Group

Use this command to change the synchronization settings and warning thresholds that the controller owner of the primary side of the asynchronous mirror group uses when it performs an initial synchronization or resynchronization. Changing the synchronization settings affects the synchronization operations of all mirrored pairs within the asynchronous mirror group.

Syntax

```
set asyncMirrorGroup ["asyncMirrorGroupName"]
[syncInterval=integer (minutes | hours | days)
warningSyncThreshold=integer (minutes | hours | days)
warningRecoveryThreshold=integer (minutes | hours | days)
warningThresholdPercent=percentValue
userLabel="New_asyncMirrorGroupName"
autoResync=(TRUE | FALSE)
volume="repos_xxxx" increaseRepositoryCapacity
(repositoryVolume=("repos_xxxx") |
repositoryVolumes=("repos_xxxx" "repos_xxxx"))
role=(primary | secondary)
(force=TRUE | FALSE | noSync=TRUE | FALSE)
```

Set Consistency Group Attributes

This command defines the properties for a snapshot consistency group.

Syntax

```
set consistencyGroup ["consistencyGroupName"]
[userLabel="consistencyGroupName" |
repositoryFullPolicy=(failBaseWrites | purgeSnapImages)|
repositoryFullLimit=percentValue|
autoDeleteLimit=numberOfSnapImages|
rollbackPriority=(lowest | low | medium | high | highest)]
```

Set Consistency Group Snapshot Volume

This command creates a unique name for a snapshot volume of a consistency group.

Syntax

```
set cgSnapVolume [cgSnapVolumeName]
userLabel="cgSnapVolumeName"
```

Set Controller

This command defines the attributes for the controllers.

Syntax

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

Set Controller Service Action Allowed Indicator

This command turns on or turns off the Service Action Allowed indicator light on a controller in a controller tray or a controller-drive tray. 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 canister or the interconnect-battery canister, use the `set tray serviceAllowedIndicator` command.)

Syntax

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

Set Disk Pool

This command sets the attributes associated with a disk pool based on the specified parameters.

Syntax

```
set diskPool (diskPool=[diskPoolName] |
diskPools=["diskPoolName1" ... "diskPoolNameN"] |
allDiskPools)
[reservedDriveCount=reservedDriveCountValue |
warningThreshold=(warningThresholdValue | default) |
criticalThreshold=(criticalThresholdValue | default) |
criticalPriority=(highest|high|medium|low|lowest) |
degradedPriority=(highest|high|medium|low|lowest)
backgroundPriority=(highest|high|medium|low|lowest) |
userLabel=diskPoolName]
```

Set Disk Pool (Modify Disk Pool)

This command adds capacity to a disk pool or changes the controller ownership for the entire disk pool. These two operations are mutually exclusive.

Syntax

```
set diskPool [diskPoolName]
((addDrives=[trayID1,drawerID1,slotID1 ...
trayIDn,drawerIDn,slotIDn] |
addCapacity=(diskPoolCapacity)) | owner=(a | b))
```

Set Drawer Service Action Allowed Indicator

This command turns on or turns off the Service Action Allowed indicator light on a drawer that holds drives. Drawers are used in high-capacity drive trays. The drawers slide out of the drive tray to provide access to the drives. Use this command only for drive trays that use drawers. 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 tray [trayID] drawer [drawerID]
serviceAllowedIndicator=(on | off | forceOnWarning)
```

Set Drive Channel Status

This command defines how the drive channel performs.

Syntax

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

Set Drive Hot Spare

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

Syntax

```
set (drive [trayID,drawerID,slotID] |
drives [trayID1,drawerID1,slotID1
... trayIDn,drawerIDn,slotIDn])
hotSpare=(TRUE | FALSE)
```

Set Drive Service Action Allowed Indicator

This command turns on or turns off the Service Action Allowed indicator light on a drive in drive trays 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 [trayID,drawerID,slotID] |
drives [trayID1,drawerID1,slotID1
... trayIDn,drawerIDn,slotIDn])
serviceAllowedIndicator=(on | off)
```

Set Drive State

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

Syntax

```
set drive [trayID,drawerID,slotID]
operationalState=failed
```

Set Foreign Drive to Native

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

Run this command to add the missing (foreign) drives back into their original volume group and to make them part of the volume groupin 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 volume group.

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 [trayID,drawerID,slotID] |
drives [trayID1,drawerID1,slotID1
... trayIDn,drawerIDn,slotIDn] |
allDrives) nativeState
```

Set Host

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

Syntax

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

Set Host Channel

This command defines the loop ID for the host channel.

Syntax

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

Set Host Group

This command renames a host group.

Syntax

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

Set Host Port

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

Syntax

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

Set iSCSI Initiator

This command sets the attributes for an iSCSI initiator.

Syntax

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

Set iSCSI Target Properties

This command defines properties for an iSCSI target.

Syntax

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

Set Read-Only Snapshot Volume to Read/Write Volume

This command changes a snapshot volume that is a read-only volume to a snapshot volume that is read/write volume. You also can use this command to identify a new repository volume for the read/write volume, or to set a full limit warning level for the repository volume.

Syntax

```
set snapVolume ["snapImageVolumeName"] convertToReadWrite
[(repositoryVolume="repos_xxxx" |
repositoryVolume=(volumeGroupName [capacity=capacityValue])
repositoryVolume=(diskPoolName [capacity=capacityValue]))
repositoryFullLimit=percentValue]
```

Set Session

This command defines how you want the current script engine session to run. For the purpose of this command a session is the duration for the running of the commands. This command does not permanently set the parameters for the storage array.

Syntax

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

Set Snapshot (Legacy) Volume

This command defines the properties for a snapshot (legacy) volume and lets you rename a snapshot (legacy) 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)
```

Set Snapshot Group Attributes

This command defines the properties for a snapshot group.

Syntax

```
set snapGroup ["snapGroupName"]
[userLabel="snapGroupName" |
repositoryFullPolicy=(failBaseWrites | purgeSnapImages) |
repositoryFullLimit=percentValue |
autoDeleteLimit=numberOfSnapImages |
rollbackPriority=(lowest | low | medium | high | highest)]
```

Set Snapshot Group Media Scan

This command runs a media scan on a snapshot group.

Syntax

```
set snapGroup ["snapGroupName"]
mediaScanEnabled=(TRUE | FALSE)
redundancyCheckEnabled=(TRUE | FALSE)
```

Set Snapshot Group Repository Volume Capacity

This command increases or decreases the capacity of a snapshot group repository volume.

Syntax for Increasing Capacity

```
set snapGroup ["snapGroupName"] increaseRepositoryCapacity
(repositoryVolumes="repos_xxxx" |
repositoryVolumes=((volumeGroupName
[capacity=capacityValue])) |
repositoryVolumes=((diskPoolName
[capacity=capacityValue])))
```

Syntax for Decreasing Capacity

```
set snapGroup ["snapGroupName"]
decreaseRepositoryCapacity
count=numberOfVolumes
```

Set Snapshot Group Schedule

This command defines the the schedule for taking snapshot images for a snapshot group.

Syntax

```
set snapGroup ["snapGroupName"]
enableSchedule=(TRUE | FALSE)
schedule (immediate | snapshotSchedule)
```

Set Snapshot Volume Media Scan

This command runs a media scan on the drives used for a snapshot volume. Optionally, you also can perform a redundancy check on the data.

Syntax

```
set snapVolume ["snapVolumeName"]
mediaScanEnabled=(TRUE | FALSE)
[redundancyCheckEnabled=(TRUE | FALSE)]
```

Set Snapshot Volume Repository Volume Capacity

This command increases or decreases the capacity of a snapshot volume repository volume.

Syntax for Increasing Capacity

```
set snapVolume[“snapVolumeName”] increaseRepositoryCapacity
repositoryVolumes=("repos_xxxx" |
repositoryVolumes=(volumeGroupName
[capacity=capacityValue]) repositoryVolumes=(diskPoolName
[capacity=capacityValue]))
```

Syntax for Decreasing Capacity

```
set snapVolume ["snapVolumeName"] decreaseRepositoryCapacity
count=numberOfVolumes
```

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" |
userRole=(admin | monitor)
userLabel="storageArrayName"
isnsRegistration=(TRUE | FALSE))
```

Set Storage Array AutoSupport Bundle Disable

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

Syntax

```
set storageArray autoSupport disable
```

Set Storage Array AutoSupport Bundle Enable

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

Syntax

```
set storageArray autoSupport enable
```

Set Storage Array ICMP Response

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

Syntax

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

Set Storage Array iSNS Server IPv4 Address

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

Syntax

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

Set Storage Array iSNS Server IPv6 Address

This command sets the IPv6 address for the iSNS server.

Syntax

```
set storageArray isnsIPv6Address=ipAddress
```

Set Storage Array iSNS Server Listening Port

This command sets the iSNS server listening port.

Syntax

```
set storageArray isnsListeningPort=listeningPortIPAddress
```

Set Storage Array iSNS Server Refresh

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

Syntax

```
set storageArray isnsServerRefresh
```

Set Storage Array Learn Cycle

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

Syntax

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

Set Storage Array Redundancy Mode

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

Syntax

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

Set Storage Array 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 volume group or disk pool, that drive will be security-enabled using the security key. Before you can set the security key, you must use the `create storageArray securityKey` command to create the security key.

Syntax

```
set storageArray securityKey
```

Set Storage Array Time

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

Syntax

```
set storageArray time
```

Set Storage Array Tray Positions

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

Syntax

```
set storageArray trayPositions=(controller | trayID ...
trayIDn)
```

Set Storage Array Unnamed Discovery Session

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

Syntax

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

Set Synchronous Mirroring

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

NOTE In previous versions of this command the feature identifier was `remoteMirror`. This feature identifier is no longer valid and is replaced by `syncMirror`.

Syntax

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

Set Thin Volume Attributes

This command defines the properties for a thin volume. You can use the parameters to define properties for one or more thin volumes.

Syntax

```
set (volume ["volumeName"] |
volumes ["volumeName1" ... "volumeNameN"] | volume <wwID>)
[newCapacity=capacityValue |
repositoryMaxCapacity=capacityValue|
repositoryExpansionPolicy=(automatic|manual)|
warningThresholdPercent=warningThresholdPercentValue |
addRepositoryCapacity=capacity-spec]
```

Set Tray Alarm

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

Syntax

```
set (allTrays | tray [trayID]
alarm=(enable | disable | mute))
```

Set Tray Identification

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

Syntax

```
set tray ["serialNumber"] id=trayID
```

Set Tray Service Action Allowed Indicator

This command turns on or turns off the Service Action Allowed indicator light on a power-fan canister, an interconnect-battery canister, or an environmental services module (ESM) canister. 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 canister, use the `set controller serviceAllowedIndicator` command.

Syntax

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

Set Volume Attributes for a Volume Group

This command defines the properties for a volume in a volume group. 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.

NOTE In configurations where volume groups 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)
dataAssuranceDisabled=(TRUE | FALSE)
```

Syntax Applicable to Only One Volume

```
set (volume ["volumeName"] | volume <wwID>)
addCapacity=volumeCapacity
[addDrives=(trayID1,drawerID1,slotID1
... trayIDn,drawerIDn,slotIDn)]
redundancyCheckEnabled=(TRUE | FALSE)
segmentSize=segmentSizeValue
userLabel=volumeName
preReadRedundancyCheck=(TRUE | FALSE)
```

Set Volume Attributes for a Disk Pool

This command defines the properties for a volume in a disk pool.

NOTE In configurations where disk 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, bring the host to a quiescent state and try the operation again.

Syntax

```
set (allVolumes | volume ["volumeName"] |
volumes ["volumeName1" ... "volumeNameN"] | volume <wwID>)
[(addCapacity = capacityValue[KB|MB|GB|TB|Bytes] |
[addDrives = (drive-spec-list)])|
cacheFlushModifier=cacheFlushModifierValue |
cacheReadPrefetch = (TRUE | FALSE) |
cacheWithoutBatteryEnabled=(TRUE | FALSE) |
mediaScanEnabled=(TRUE | FALSE) |
mirrorCacheEnabled=(TRUE | FALSE) |
owner=(a | b) |
preReadRedundancyCheck = (TRUE | FALSE) |
readCacheEnabled=(TRUE | FALSE) |
redundancyCheckEnabled = (TRUE | FALSE) |
segmentSize = segmentSizeValue
userLabel = userlabelValue
writeCacheEnabled=(TRUE | FALSE) |
dataAssuranceDisabled=(TRUE | FALSE)]
```

Set Volume Copy

NOTE With firmware version 7.83 the `copyType=(online | offline)` parameter is no longer used.

This command defines the properties for a volume copy pair. This command is valid for both snapshot (legacy) volume copy pairs and new snapshot volume copy pairs.

Syntax

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

Set Volume Group

This command defines the properties for a volume group.

Syntax

```
set volumeGroup [volumeGroupName]
addDrives=(trayID1,drawerID1,slotID1
... trayIDn,drawerIDn,slotIDn)
raidLevel=(0 | 1 | 3 | 5 | 6)
owner=(a | b)
```

Set Volume Group Forced State

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

Syntax

```
set volumeGroup [volumeGroupName] forcedState
```

Set Volume Mapping

This command defines the logical unit number (LUN) mapping between a volume a host or host group. This command is applicable to volumes in either a volume group or disk pools.

NOTE You cannot use this command for a snapshot volume that is used in online volume copy.

Syntax

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

Show Alert Severities

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

This command shows all of the severities for which an alert is sent. This command cannot show information for a specific type of severity.

Syntax

```
SMcli -alertSeverities
```

Show Asynchronous Mirror Group

This command displays configuration information for one or more asynchronous mirror groups. This command also displays the asynchronous mirrored pairs associated with each asynchronous mirror group, including incomplete asynchronous mirrored pairs.

You alsocan use this command to show the progress of periodic data synchronization on all of the mirrored pairs within the asynchronous mirror group.

Syntax

```
show (allAsyncMirrorGroups | asyncMirrorGroup
["asyncMirrorGroupName"])
[summary]
```

Show Asynchronous Mirror Group Synchronization Progress

This command displays the progress of *periodic* synchronization of the asynchronous mirror group between the local and remote storage array. This command returns the progress of data synchronization on all of the mirrored pairs within the asynchronous mirror group. This command shows the progress as a percentage of data synchronization that has been completed.

NOTE There are two types of synchronization: initial synchronization and periodic synchronization. Initial asynchronous mirror group synchronization progress is displayed in the **Long Running Operations** dialog and by executing the `show storageArray longRunningOperations` command.

Syntax

```
show asyncMirrorGroup ["asyncMirrorGroupName"]
[synchronizationProgress]
```

Show Cache Backup Device Diagnostic Status

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

Syntax

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

Show Cache Memory Diagnostic Status

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

Syntax

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

Show Consistency Group

This command returns information about one or more consistency groups.

Syntax

```
show (consistencyGroup [consistencyGroupName] |
consistencyGroups [consistencyGroupName1 ...
consistencyGroupNameN] |
allConsistencyGroups)
[(summary | schedule)]
```

Show Consistency Group Snapshot Image

This command shows one snapshot image or several snapshot images that are in one or more snapshot consistency groups.

Syntax

```
show ((CGSnapImage [(CGSnapImageName |
CGSnapImageSequenceNumber)]) |
CGSnapImages [(CGSnapImageNumber1
... CGSnapImageNumbern |
CGSnapImageSequenceNumber1
... CGSnapImageSequenceNumbern)]) |
allCGSnapImages)
[summary]
```

Show Controller

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

- The status (Online or Offline)
- The current firmware and NVSRAM configuration

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

Syntax

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

Show Controller Diagnostic Status

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

Syntax

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

Show Controller NVSRAM

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. To view an example of a table of NVSRAM values that are returned by this command, refer to the "Examples of Information Returned by the Show Command" appendix in Configuring and Maintaining a Storage Array Using the Command Line Interface document.

Syntax

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

Show Current iSCSI Sessions

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

Syntax

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

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

Show Disk Pool

This command returns this information about a disk pool:

- The status (such as optimal, online, offline)
- The total capacity
- The preservation capacity, usable capacity and unusable capacity
- The current owner (the controller in slot A or the controller in slot B)
- The drive media (Fibre Channel, SATA, or SAS)
- The drive interface (Fibre Channel, SATA, or SAS)
- The associated volumes and free capacity
- The associated drives

Syntax

```
show diskPool [diskPoolName]
```

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 tray 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 tray location and the slot location
  - The preferred channel
  - The redundant channel
- Hot spare coverage
- Details for each drive

Depending on the size of your storage array, this information can be several pages long. To view an example of the drive information that is returned by the `show drives` command, refer to the "Examples of Information Returned by the Show Commands" topic in "Configuring and Maintaining a Storage Array Using the Command Line Interface." In addition, the drive information is returned for the `show storageArray profile` command.

Syntax

```
show (allDrives
[driveMediaType=(HDD | SSD | unknown | allMedia)] |
[driveType=(fibre | SATA | SAS)]) |
drive [trayID,drawerID,slotID] |
drives [trayID1,drawerID1,slotID1
... trayIDn,drawerIDn,slotIDn])
summary
```

Show Drive Channel Statistics

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

Syntax

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

Show Drive Download Progress

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

Syntax

```
show allDrives downloadProgress
```

Show Drive Performance Statistics

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

- The storage array in which has the drive or drives
- The current I/O latency
- The maximum I/O latency
- The minimum I/O latency
- The average I/O latency

Syntax

```
show (allDrives
drive [trayID,drawerID,slotID] |
drives [trayID1,drawerID1,slotID1
... trayIDn,drawerIDn,slotIDn])
performanceStats
```

Show Host Interface Card Diagnostic Status

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

Syntax

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

Show Host Ports

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

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

Syntax

```
show allHostPorts
```

Show Snapshot Group

This command returns this information about one or more snapshot image groups.

Syntax

```
show (allSnapGroups | snapGroup [snapGroupName] |
snapGroups ["snapGroupName1" ... "snapGroupNamen"])
[summary | schedule]
```

Show Snapshot Image

This command returns information about the snapshot images that a user had previously created.

Syntax for Showing a Specific Snapshot Image

```
show (snapImage ["snapImageName"] |
snapImages ["snapImageName1" ... "snapImageNamen"] |
allSnapImages])
[summary]
```

Show Snapshot Volumes

This command returns information about one or more snapshot volumes.

Syntax

```
show (allSnapVolumes | snapVolume ["snapVolumeName"] |
snapVolumes ["snapVolumeName1" ... "snapVolumeNamen"])
[summary]
```

Show SSD Cache

This command displays information about the SSD cache.

Syntax

```
show ssdCache [ssdCacheName]
```

Show SSD Cache Statistics

This command displays data about the SSD cache usage.

Syntax

```
show ssdCache [ssdCacheName] ssdCacheStatistics
[controller=[(a|b|both)]
file="filename"]
```

Show Storage Array

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

Syntax

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

Show Storage Array Auto Configure

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

Syntax

```
show storageArray autoConfiguration
[driveType=(fibre | SATA | SAS)
driveMediaType=(HDD | SSD | unknown | allMedia)
raidLevel=(0 | 1 | 3 | 5 | 6)
volumeGroupWidth=numberOfDrives
volumeGroupCount=numberOfVolumeGroups
volumesPerGroupCount=numberOfVolumesPerGroup
hotSpareCount=numberOfHotspares
segmentSize=segmentSizeValue
cacheReadPrefetch=(TRUE | FALSE)
securityType=(none | capable | enabled)]
```

Show Storage Array Core Dump

This command show details of the core dump on the controller cache, if a core dump is available.

Syntax

```
show storageArray coreDump
```

Show Storage Array DBM Database

This command retrieves and shows metadata for the on-board backup locations of a storage array. When there are multiple backup locations, metadata is shown for each location.

Syntax

```
show storageArray dbmDatabase
```

Show Storage Array Host Topology

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

Syntax

```
show storageArray hostTopology
```

Show Storage Array LUN Mappings

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

Syntax

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

Show Storage Array Negotiation Defaults

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

Syntax

```
show storageArray  iscsiNegotiationDefaults
```

Show Storage Array Unconfigured iSCSI Initiators

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

Syntax

```
show storageArray unconfiguredIscsiInitiators
```

Show Storage Array Unreadable Sectors

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

- Volume user label
- Logical unit number (LUN)

- Accessible by (host or host group)
- Date/time
- Volume-relative logical block address (hexadecimal format – 0xnnnnnnnn)
- Drive location (tray t, slot s)
- Drive-relative logical block address (hexadecimal format – 0xnnnnnnnn)
- Failure type

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

Syntax

```
show storageArray unreadableSectors
```

Show String

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

Syntax

```
show "textString"
```

Show Synchronous Mirroring Volume Candidates

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

NOTE In previous versions of this command the feature identifier was `remoteMirror`. This feature identifier is no longer valid and is replaced by `syncMirror`.

Syntax

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

Show Synchronous Mirroring Volume Synchronization Progress

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

NOTE In previous versions of this command the feature identifier was `remoteMirror`. This feature identifier is no longer valid and is replaced by `syncMirror`.

Syntax

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

Show Thin Volume

This command returns the expansion history or the consumed capacity for the specified thin volume or volumes.

Syntax

```
show (allVolumes | volume [volumeName] |
volumes ["volumeName1" ... "volumeNameN"]) (consumedCapacity
| (expansionHistory [file=fileName]))
```

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 volume group where the volume is located
- Details:
  - The volume ID
  - The subsystem ID
  - The drive type (Fibre Channel, SATA, or SAS)
  - Tray 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 (legacy) repository volumes
- The mirror repository volumes

- The snapshot (legacy) volumes
- The snapshot (legacy) copies

To view an example of the information returned by this command, refer to the topic "Examples of Information Returned by the Show Commands" in *Configuring and Maintaining a Storage Array Using the Command Line Interface*.

Syntax

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

Show Volume Action Progress

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

For a long-running operation that is currently running on a volume, this command returns information about the volume action and amount of the long-running operation that is completed. The amount of the long-running operation that is completed is shown as a percentage (for example, 25 means that 25 percent of the long-running operation is completed).

Syntax

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

Show Volume Copy

This command returns this information about volume copy operations:

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

You can retrieve information about a specific volume copy pair or all of the volume copy pairs in the storage array. This command is valid for both snapshot (legacy) volume copy pairs and new snapshot volume copy pairs.

Syntax

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

Show Volume Copy Source Candidates

This command returns information about the candidate volumes that you can use as the source for a volume copy operation. This command is valid for both snapshot (legacy) volume copy pairs and new snapshot volume copy pairs.

Syntax

```
show volumeCopy sourceCandidates
```

Show Volume Copy Target Candidates

This command returns information about the candidate volumes that you can use as the target for a volume copy operation. This command is valid for both snapshot (legacy) volume copy pairs and new snapshot volume copy pairs.

Syntax

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

Show Volume Group

This command returns this information about a volume group:

- The status (Online or Offline)
- The drive type (Fibre Channel, SATA, or SAS)
- Tray 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 volumeGroup [volumeGroupName]
```

Show Volume Group Export Dependencies

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

Syntax

```
show volumeGroup [volumeGroupName] exportDependencies
```

Show Volume Group Import Dependencies

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

Syntax

```
show volumeGroup [volumeGroupName] importDependencies
[cancelImport=(TRUE | FALSE)]
```

Show Volume Performance Statistics

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

- Storage Arrays
- Total IOs/s
- Read %
- Primary Read Cache Hit %
- Primary Write Cache Hit %
- SSD Cache Hit %
- Current MBs/s
- Maximum MBs/s
- Current IOs/s
- Maximum IOs/s
- Minimum IOs/s
- Average IOs/s
- Minimum MBs/s
- Average MBs/s
- Current IO Latency
- Maximum IO Latency
- Minimum IO Latency
- Average IO Latency

Syntax

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

Show Volume Reservations

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

Syntax

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

Start Asynchronous Mirroring Synchronization

This command starts Asynchronous Mirroring synchronization.

Syntax

```
start asyncMirrorGroup ["asyncMirrorGroupName"] synchronize
```

Start Cache Backup Device Diagnostic

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

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

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

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

Syntax

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

Start Cache Memory Diagnostic

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

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

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

Syntax

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

Start Consistency Group Snapshot Rollback

This command starts a rollback operation to the member base volumes in a snapshot consistency group. The content of the base volumes changes immediately to match the point-in-time content of the consistency group snapshot volume. The base volumes immediately becomes available for read/write requests after the rollback operation has successfully completed.

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

To stop a rollback operation to the member base volumes use the `stop cgSnapImage rollback` command.

Syntax

```
start cgSnapImage ["snapCGID:imageID"] rollback
[memberVolumeSet ["memberVolumeName1"
... "memberVolumeNameN"]]
```

Start Controller Diagnostic

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

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

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

Syntax

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

Start Controller Trace

This command starts an operation that saves debug trace information to a compressed file. The debug trace information can be used by a Technical Support Representative to help analyze how well a storage array is running.

Syntax

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

Start Disk Pool Locate

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

Syntax

```
start diskPool [diskPoolName] locate
```

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=[trayID1 (left | right), ... , trayIDN (left | right)] |
drives=[trayID1,slotID1, ... , trayIDn,slotIDN]] |
dataPattern=(fixed | pseudoRandom) |
patternNumber=[(0xhexadecimal | number)] |
maxErrorCount=integer |
testIterations=integer |
timeout=timeInterval)
```

Start Drive Channel Locate

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

Syntax

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

Start Drive Initialize

This command starts drive initialization.

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

Syntax

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

Start Drive Locate

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

Syntax

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

Start Drive Reconstruction

This command starts reconstructing a drive.

Syntax

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

Start Host Interface Card Diagnostic

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

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

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

Syntax

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

Start iSCSI DHCP Refresh

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

Syntax

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

Start Secure Drive Erase

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

Syntax

```
start secureErase (drive [trayID,slotID] |
drives [trayID1,slotID1 ... trayIDn,slotIDn])
```

Start Snapshot (Legacy) Rollback

NOTE With firmware version 7.83 the order of the terms in the syntax is changed to be consistent with other commands. Replace the deprecated command syntax with the new comman syntax.

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

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

Syntax (Deprecated)

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

Syntax (New)

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

Start Snapshot Image Rollback

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

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

NOTE You cannot use this command for snapshot images involved in online volume copy.

Syntax

```
start snapImage ["snapImageName"] rollback
```

Start SSD Cache Performance Modeling

This command starts performance modeling for the SSD cache. Performance modeling monitors and measures I/O activity for a period of time and estimates performance for various SSD cache sizes. Performance is estimated using two metrics: cache hit percentage and average response time. The performance modeling data is not available until you stop the operation using the `stop ssdCache performanceModeling` command.

Syntax

```
start ssdCache [ssdCacheName] performanceModeling
```

Start Storage Array Configuration Database Diagnostic

This command runs a consistency check against a configuration database.

Syntax

```
start storageArray configDbDiagnostic
[sourceLocation=(disk | onboard) |
diagnosticType=(fileSystem | mirror) |
controller[(a | b)]]
```

Start Storage Array Core Dump

This command produces a storage array core dump.

Syntax

```
start storageArray coreDump controller [(a|b)]
```

Start Storage Array iSNS Server Refresh

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

NOTE This command is for IPv4 only.

Syntax

```
start storageArray isnsServerRefresh
```

Start Storage Array Locate

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

Syntax

```
start storageArray locate
```

Start Synchronous Mirroring Synchronization

This command starts Synchronous Mirroring synchronization.

NOTE In previous versions of this command the feature identifier was `remoteMirror`. This feature identifier is no longer valid and is replaced by `syncMirror`.

Syntax

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

Start Tray Locate

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

Syntax

```
start tray [trayID] locate
```

Start Volume Group Defragment

This command starts a defragment operation on the specified volume group.

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

Syntax

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

Start Volume Group Export

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

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

Syntax

```
start volumeGroup [volumeGroupName] export
```

Start Volume Group Import

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

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

Syntax

```
start volumeGroup ["volumeGroupName"] import
```

Start Volume Group Locate

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

Syntax

```
start volumeGroup [volumeGroupName] locate
```

Start Volume Initialization

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

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

Syntax

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

Stop Cache Backup Device Diagnostic

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

Syntax

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

Stop Cache Memory Diagnostic

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

Syntax

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

Stop Consistency Group Snapshot Rollback

This command stops a rollback operation to the member base volumes in a snapshot consistency group that was initiated by the `start cgSnapImage rollback` command.

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

Syntax

```
stop cgSnapImage["snapCGID:imageID"]| rollback
[memberVolumeSet ["memberVolumeName1" ...
"memberVolumeNamen"]]
```

Stop Consistency Group Snapshot Volume

This command stops the copy-on-write operation for creating a consistency group snapshot volume. To restart the copy-on-write operation use the `resume cgSnapVolume` command.

NOTE This command does not delete a consistency group snapshot volume. To delete a consistency group snapshot volume use the `delete cgSnapVolume` command.

Syntax

```
stop cgSnapVolume [snapVolumeName]
```

Stop Controller Diagnostic

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

Syntax

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

Stop Disk Pool Locate

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

Syntax

```
stop diskPool locate
```

Stop Drive Channel Fault Isolation Diagnostics

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

Syntax

```
stop driveChannel faultDiagnostics
```

Stop Drive Channel Locate

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

Syntax

```
stop driveChannel locate
```

Stop Drive Locate

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

Syntax

```
stop drive locate
```

Stop Host Interface Card Diagnostic

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

Syntax

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

Stop Pending Snapshot Images on Consistency Group

This command stops all of the pending snapshot images that are to be created on a snapshot consistency group. If the snapshot consistency group does not have any pending snapshot images, the storage management software displays an error message and does not run the command.

Syntax

```
stop consistencyGroup [consistencyGroupName]
pendingSnapImageCreation
```

Stop Snapshot Group Pending Snapshot Images

This command cancels all of the pending snapshot images that are to be created on a snapshot group. If the snapshot group does not have any pending snapshot images, the firmware displays an error message and does not run the command. You can run this command on a snapshot group or a snapshot consistency group.

Syntax

```
stop (snapGroup [snapGroupName] |
consistencyGroup [snapConsistencyGroupName])
pendingSnapImageCreation
```

Stop Snapshot (Legacy)

This command stops a copy-on-write operation.

Syntax

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

Stop Snapshot (Legacy) Rollback

This command stops a snapshot (legacy) rollback operation that was initiated by 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 (legacy) volume becomes disabled and unusable.

Syntax

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

Stop Snapshot Image Rollback

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

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

Syntax

```
stop snapImage [snapCGID:imageID] rollback
```

Stop Snapshot Volume

This command stops a snapshot volume operation.

Syntax

```
stop snapVolume ["snapVolumeName"]
```

Stop SSD Cache Performance Modeling

This command stops the performance modeling operation and displays the performance modeling data for the SSD cache. Before running this command, you must start the performance modeling operation with the `start ssdCache performanceModeling` command. Optionally, you can save the data to a file.

Syntax

```
stop ssdCache [ssdCacheName] performanceModeling
[file="filename"]
```

Stop Storage Array Configuration Database Diagnostic

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

Syntax

```
stop storageArray configDbDiagnostic
```

Stop Storage Array Drive Firmware Download

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

Syntax

```
stop storageArray driveFirmwareDownload
```

Stop Storage Array iSCSI Session

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

Syntax

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

Stop Storage Array Locate

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

Syntax

```
stop storageArray locate
```

Stop Tray Locate

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

Syntax

```
stop tray locate
```

Stop Volume Copy

This command stops a volume copy operation. This command is valid for both snapshot (legacy) volume copy pairs and new snapshot volume copy pairs.

Syntax

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

Stop Volume Group Locate

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

Syntax

```
stop volumeGroup locate
```

Suspend Asynchronous Mirror Group

This command suspends the synchronization of data on all mirrored pairs at the asynchronous mirror group level. This `suspend` command helps to reduce any performance impact to the host application that might occur while any changed data on the local storage array is copied to the remote storage array.

Syntax

```
suspend asyncMirrorGroup ["asyncMirrorGroupName"]
```

Suspend SSD Cache

This command temporarily stops caching for all of the volumes that are using the SSD cache. While caching is stopped, host reads are serviced from the base volumes instead of from the SSD cache.

Syntax

```
suspend ssdCache [ssdCacheName]
```

Suspend Synchronous Mirroring

This command suspends a Synchronous Mirroring operation.

NOTE In previous versions of this command the feature identifier was `remoteMirror`. This feature identifier is no longer valid and is replaced by `syncMirror`.

Syntax

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

Test Alert Severities

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

---

This command sends out a test alert to the Windows Event Log and all configured syslog receivers.

Syntax

```
SMcli -alertTest
```

Test Asynchronous Mirror Group Connectivity

This command tests possible communication problems between the local storage array and the remote the storage array associated with an asynchronous mirror group.

Syntax

```
diagnose asyncMirrorGroup ["asyncMirrorGroupName"]
testID=(all | connectivity | latency | bandwidth |
portConnections)
```

Validate Storage Array Security Key

This command validates the security key for a storage array that has full disk encryption (FDE) drives to make sure that the security key is not corrupt.

Syntax

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

Command Reference - Listed by Function

Asynchronous Mirroring Commands

Activate Asynchronous Mirroring

Add Volume to Asynchronous Mirror Group

Cancel Asynchronous Mirror Group Role Reversal

Check Asynchronous Mirror Group Consistency

Clear Asynchronous Mirroring Fault

Check Storage Array Connectivity

Create Asynchronous Mirror Group

Deactivate Asynchronous Mirroring

Delete Asynchronous Mirror Group

Establish Asynchronous Mirrored Pair

Remove Incomplete Asynchronous Mirrored Pair from Asynchronous Mirror Group

Remove Volume from Asynchronous Mirror Group

Reset Asynchronous Mirror Group Statistics

Reset iSCSI IP Address

Resume Asynchronous Mirror Group

Save Asynchronous Mirror Group Statistics

Set Asynchronous Mirror Group

Show Asynchronous Mirror Group

Show Asynchronous Mirror Group Synchronization Progress

Start Asynchronous Mirroring Synchronization

Suspend Asynchronous Mirror Group

Test Asynchronous Mirror Group Connectivity

AutoSupport Bundle Collection Commands

Disable AutoSupport at the EMW Level SMcli Version

Enable AutoSupport at the EMW Level SMcli Version

Set Storage Array AutoSupport Bundle Disable

Set Storage Array AutoSupport Bundle Enable

Consistency Group Commands

- Add Member to Consistency Group
- Create Consistency Group
- Create Consistency Group Snapshot Image
- Create Consistency Group Snapshot Volume
- Create Consistency Group Snapshot Volume Mapping
- Delete Consistency Group
- Delete Consistency Group Snapshot Image
- Delete Consistency Group Snapshot Volume
- Remove Member Volume from Consistency Group
- Resume Consistency Group Snapshot Volume
- Set Consistency Group Attributes
- Set Consistency Group Snapshot Volume
- Show Consistency Group
- Show Consistency Group Snapshot Image
- Start Consistency Group Snapshot Rollback
- Stop Consistency Group Snapshot Rollback
- Stop Consistency Group Snapshot Volume
- Stop Pending Snapshot Images on Consistency Group

Controller Commands

- Clear Drive Channel Statistics
- Diagnose Controller
- 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
- Show Cache Backup Device Diagnostic Status
- Show Cache Memory Diagnostic Status

Show Controller

Show Controller Diagnostic Status

Show Controller NVSRAM

Show Drive Channel Statistics

Show Host Interface Card Diagnostic Status

Start Cache Backup Device Diagnostic

Start Cache Memory Diagnostic

Start Controller Diagnostic

Start Controller Trace

Start Drive Channel Fault Isolation Diagnostics

Start Drive Channel Locate

Start Host Interface Card Diagnostic

Stop Cache Backup Device Diagnostic

Stop Cache Memory Diagnostic

Stop Controller Diagnostic

Stop Drive Channel Fault Isolation Diagnostics

Stop Drive Channel Locate

Stop Host Interface Card Diagnostic

Core Dump Commands

Clear Storage Array Core Dump

Save Storage Array Core Dump

Start Storage Array Core Dump

Database Commands

Clear Storage Array Configuration

Clear Storage Array Recovery Mode

Save Storage Array DBM Database

Save Storage Array DBM Validator Information File

Show Storage Array DBM Database

Start Storage Array Configuration Database Diagnostic

Stop Storage Array Configuration Database Diagnostic

Disk Pool Commands

Create Disk Pool

Delete Disk Pool

Set Disk Pool

Set Disk Pool (Modify Disk Pool)

Show Disk Pool

Start Disk Pool Locate

Stop Disk Pool Locate

Drive Commands

Download Drive Firmware

Replace Drive

Revive Drive

Save Drive Channel Fault Isolation Diagnostic Status

Save Drive Log

Set Drive Hot Spare

Set Drive Service Action Allowed Indicator

Set Drive State

Set Foreign Drive to Native

Show Drive

Show Drive Download Progress

Start Drive Channel Fault Isolation Diagnostics

Start Drive Initialize

Start Drive Locate

Start Drive Reconstruction

Start Secure Drive Erase

Stop Drive Channel Fault Isolation Diagnostics

Stop Drive Locate

Feature Management Commands

Enable Storage Array Feature

Disable Storage Array Feature

Host Topology Commands

Activate Host Port

Activate iSCSI Initiator

Create Host

Create Host Group

Create Host Port

Create iSCSI Initiator

Delete Host

Delete Host Group

Delete Host Port

Delete iSCSI Initiator

Set Host

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

Synchronous Mirroring Commands

Activate Synchronous Mirroring

Check Synchronous Mirroring Status

Create Synchronous Mirroring

Deactivate Synchronous Mirroring

Diagnose Synchronous Mirroring

Re-create Synchronous Mirroring Repository Volume

Remove Synchronous Mirroring

Resume Synchronous Mirroring

Set Synchronous Mirroring

Show Synchronous Mirroring Volume Candidates

Show Synchronous Mirroring Volume Synchronization Progress

Start Synchronous Mirroring Synchronization

Suspend Synchronous Mirroring

Core Dump Commands

Check Repository Consistency

Session Command

Set Session

Snapshot (Legacy) Commands

Create Snapshot (Legacy) Volume

Delete Snapshot (Legacy) Volume

Re-create Snapshot (Legacy)

Resume Snapshot (Legacy) Rollback

Set Snapshot (Legacy) Volume

Start Snapshot (Legacy) Rollback

Stop Snapshot (Legacy)

Stop Snapshot (Legacy) Rollback

Snapshot Group Commands

Convert Snapshot (Legacy) Volume to Snapshot Group

Create Snapshot Group

Revive Snapshot Group

Set Snapshot Group Attributes

Set Snapshot Group Media Scan

Set Snapshot Group Repository Volume Capacity

Set Snapshot Group Schedule

Show Snapshot Group

Stop Snapshot Group Pending Snapshot Images

Snapshot Image Commands

Create Snapshot Image

Delete Snapshot Image

Show Snapshot Image

Start Snapshot Image Rollback

Stop Snapshot Image Rollback

Snapshot Volume Commands

Create Read-Only Snapshot Volume

Create Snapshot Volume

Rename Snapshot Volume

Resume Snapshot Volume

Revive Snapshot Volume

Set Read-Only Snapshot Volume to Read/Write Volume

Set Snapshot Volume Repository Volume Capacity

Set Snapshot Volume Media Scan

Show Snapshot Volumes

SSD Cache Commands

Add Drives to SSD Cache

Change SSD Cache Application Type

Create SSD Cache

Delete SSD Cache

Enable or Disable SSD Cache for a Volume

Locate SSD Cache

Remove Drives from SSD Cache

Rename SSD Cache

Resume SSD Cache

Show SSD Cache

Stop SSD Cache Performance Modeling

Start SSD Cache Performance Modeling

Suspend SSD Cache

Storage Array Commands

Activate Storage Array Firmware

Autoconfigure Storage Array

Autoconfigure Storage Array Hot Spares

Clear Storage Array Configuration

Clear Storage Array Event Log

Clear Storage Array Firmware Pending Area

Create Storage Array Security Key

Disable External Security Key Management

Disable Storage Array Feature

Download Storage Array Drive Firmware

Download Storage Array Firmware/NVSRAM

Download Storage Array NVSRAM

Enable External Security Key Management

Enable Storage Array Feature

Export Storage Array Security Key

Import Storage Array Security Key

Load Storage Array DBM Database"

Re-create External Security Key

Reset Storage Array Battery Install Date

Reset Storage Array Diagnostic Data

Reset Storage Array Infiniband Statistics Baseline

Reset Storage Array iSCSI Baseline

Reset Storage Array RLS Baseline

Reset Storage Array SAS PHY Baseline

Reset Storage Array SOC Baseline

Reset Storage Array Volume Distribution

Save Storage Array Configuration

Save Storage Array DBM Database

Save Storage Array DBM Validator Information File

Save Storage Array Diagnostic Data

Save Storage Array Events

Save Storage Array Firmware Inventory

Save Storage Array InfiniBand Statistics

Save Storage Array iSCSI Statistics

- Stop Storage Array Locate
- Validate Storage Array Security Key

Support Bundle Collection Commands

- Configure Automatic Support Bundle Collection
- Display Automatic Support Bundle Collection Configuration
- Schedule Automatic Support Bundle Collection Configuration

Tray Commands

- Download Tray Configuration Settings
- Save Tray Log
- Set Drawer Service Action Allowed Indicator
- Set Tray Alarm
- Set Tray Identification
- Set Tray Service Action Allowed Indicator
- Start Tray Locate
- Stop Tray Locate

Uncategorized Commands

- Set Storage Array ICMP Response
- Set Storage Array iSNS Server IPv4 Address
- Set Storage Array iSNS Server IPv6 Address
- Set Storage Array iSNS Server Listening Port
- Set Storage Array iSNS Server Refresh
- Set Storage Array Unnamed Discovery Session
- Show Storage Array Negotiation Defaults
- Show String

Volume Commands

- Check Volume Parity
- Clear Volume Reservations
- Clear Volume Unreadable Sectors
- Create RAID Volume (Automatic Drive Select)
- Create RAID Volume (Free Extent Based Select)
- Create RAID Volume (Manual Drive Select)
- Delete Volume
- Remove Volume LUN Mapping

Repair Volume Parity

Set Volume Attributes for a Volume Group

Show Volume

Show Volume Action Progress

Show Volume Performance Statistics

Show Volume Reservations

Start Volume Initialization

Volume Commands for Disk Pools

Create Volume in Disk Pool

Delete Volume from Disk Pool

Initialize Thin Volume

Set Volume Attributes for a Disk Pool

Set Thin Volume Attributes

Set Volume Attributes for a Volume Group

Set Volume Mapping

Show Thin Volume

Volume Copy Commands

Create Volume Copy

Recopy Volume Copy

Remove Volume Copy

Set Volume Copy

Show Volume Copy

Show Volume Copy Source Candidates

Show Volume Copy Target Candidates

Stop Volume Copy

Volume Group Commands

Create Volume Group

Delete Volume Group

Enable Volume Group Security

Revive Volume Group

Set Volume Group

Set Volume Group Forced State

Show Volume Group

Show Volume Group Export Dependencies

Show Volume Group Import Dependencies

Start Volume Group Defragment

Start Volume Group Export

Start Volume Group Import

Start Volume Group Locate

Stop Volume Group Locate