BROADCOM®

StorCLI 12Gb/s MegaRAID® Tri-Mode

User Guide
Version 1.5

Table of Contents

Supported Controllers and Operating Systems

The Broadcom StorCLI tool works on MegaRAID®, Software RAID (SWR), and Initiator-Target (IT) controllers.

Supported Controllers

The StorCLI tool supports the following controllers:

- MegaRAID 12Gb/s SAS RAID controllers
- Initiator-Target 3 12Gb/s (IT3.x) HBA controllers
- Software RAID 12Gb/s (SWR) controllers

Supported Operating Systems

The following table lists the supported operating systems.

Table 1: Supported Operating Systems

Supported Operating Systems	Version/Flavors
Microsoft	**Microsoft Windows Client versions** • Windows Client 10 • Windows 8.1 • Windows 8 **Microsoft Windows Server versions** • Windows Server 2016 • Windows Server 2012 R2 • Windows Server 2012
Linux	**Red Hat** • Red Hat Enterprise Linux 7.5 • Red Hat Enterprise Linux 7.4 • Red Hat Enterprise Linux 7.3 • Red Hat Enterprise Linux 7.2 • Red Hat Enterprise Linux 7.1 • Red Hat Enterprise Linux 7.0 • Red Hat Enterprise Linux 6.8 • Red Hat Enterprise Linux 6.7 • Red Hat Enterprise Linux 6.6 **SUSE** • SUSE Linux Enterprise Server 12 SP3 • SUSE Linux Enterprise Server 12 SP2 • SUSE Linux Enterprise Server 12 SP1 • SUSE Linux Enterprise Server 12 • SUSE Linux Enterprise Server 11 SP4 • SUSE Linux Enterprise Server 11 SP3 • SUSE Linux Enterprise Server 11 SP2

Supported Operating Systems	Version/Flavors
VMware	• VMware ESXi 6.7 • VMware ESXi 6.5 Update 1 • VMware ESXi 6.5 • VMware ESXi 6.0 Update 2 • VMware ESXi 6.0 Update 1 • VMware ESXi 6.0 • VMware ESXi 5.5 Update 3 • VMware ESXi 5.5 Update 2 • VMware ESXi 5.5 Update 1
Citrix XenServer	• Citrix XenServer 7.2 • Citrix XenServer 7 • Citrix XenServer 6.5 • Citrix XenServer 6.5 SP1
CentOS	• CentOS-7.4 • CentOS-7.2 • CentOS-7.1 • CentOS-6.8 • CentOS-6.7
Fedora	Fedora 24 Workstation
Unified Extensible Firmware Interface	UEFI environment
ARM	Windows, Linux, UEFI

Installing StorCLI on MegaRAID Controllers

The following topics detail the steps that are required to install the StorCLI tool for MegaRAID (MR) controllers on various operating systems.

> **NOTE**
>
> StorCLI default logging requires a minimum of 20 MB of free space.

Installing the StorCLI Tool on Microsoft Windows Operating Systems

The Windows StorCLI binary is provided in a binary format, and no separate installation is required.

1. Copy the binary file from the Broadcom website.
2. Place the binary file in the directory from which you want to run StorCLI, and run the tool.

   Because Windows PowerShell is not fully supported by the StorCLI tool, use either one of the following techniques to run commands in the StorCLI tool in Windows PowerShell:

   - Enclose commands in double quotation marks; for example,

     ```
     storcli "/cx show"
     ```
   - Launch the command prompt from within Windows PowerShell to run the StorCLI commands.

     > **NOTE**
     > The StorCLI tools must be run with the administrator privileges.

Installing the StorCLI Tool on the UEFI Environment

The UEFI StorCLI binary is provided in a binary format, and no separate installation is required.

> **NOTE**
>
> StorCLI default logging requires a minimum of 20 MB of free space.

1. Copy the binary file from the Broadcom website or from the CD provided to you onto a USB drive.
2. Using the USB drive, place the binary file in the directory from which you want to run the Storage Command Line Interface, and run the tool.

After the binaries are copied, you can start executing the StorCLI commands.

Installing the StorCLI Tool on Linux Operating Systems

To install the StorCLI tool on Linux operating systems, perform the following steps:

1. Unzip the StorCLI tool package.
2. To install the StorCLI RPM feature, run the `rpm -ivh <StorCLI- x.xx-x.noarch.rpm >` command.

   By default, the StorCLI tool will be installed in the `/opt/MegaRAID/storcli` location.
3. To upgrade the StorCLI RPM feature, run the `rpm -Uvh <StorCLI- x.xx-x.noarch.rpm >` command.

Uninstalling the StorCLI Tool on Linux Operating Systems

To uninstall the StorCLI tool on Linux operating systems, perform the following steps:

Enter the `rpm -e <StorCLI -x.xx-x.noarch.rpm` command.

Installing the StorCLI Tool on VMware Operating Systems

To install the StorCLI tool on VMware operating systems, run the following from the command line:

```
esxcli software vib install -v=<path-to-vib-package> --no-sig-check
```

> **NOTE**
>
> StorCLI default logging requires a minimum of 20 MB of free space.

Example:

```
esxcli software vib install -v=/vmfs/volumes/datastore1/StorCliMN/vmware-esx-StorCli-1.01.04.vib
```

> **NOTE**
>
> Broadcom provides three variants of StorCLI tool for VMware to be compatible with ESXi versions and MegaRAID (MR) drivers:
>
> **VMware-NDS** – This package must be used with MegaRAID/IT drivers, `lsi_mr3` , which is a native driver.
>
> The **VMware-NDS** package can be executed with both native and legacy drivers.

Uninstalling the StorCLI Tool on VMware Operating Systems

Perform the following step to uninstall StorCLI on VMware OS:

Enter the `esxcli software vib remove -n =<StorCLI package name>` command.

StorCLI Tool Command Syntax

This section describes the StorCLI command syntax and the valid values for each parameter in the general command syntax.

- In large configurations, running two instances of the StorCLI tool in parallel (at the same time) is not recommended.
- To get the output in JSON format, add J at the end of the command syntax. For example:
  ```
  storcli /cx show <property1>|<property2> J
  ```
- Background operations are blocked in the UEFI environment, and these operations are resumed in the operating system environment.
- StorCLI discovers MegaRAID controllers in a driver-less UEFI environment only if the firmware is in operational or ready state.
- On ARM-UEFI platforms, StorCLI does not detect the controller if there is no ARM-UEFI driver present.

The StorCLI tool syntax uses the following general format:

```
<[object identifier]> <verb> <[adverb | attributes | properties]> <[key=value]>
```

The StorCLI tool supports the object identifiers listed in the following table.

Table 2: Object Identifiers in the StorCLI Command Syntax

Object Identifier	Description
No object identifier specified	If no object identifier exists, the command is a system command.
/cx	This object identifier is for controller x.
/call	This object identifier is for sending the command to all controllers.

Object Identifier	Description
/c *x*/v *x*	This object identifier is for a virtual drive *x* on controller *x*.
/c *x*/vall	This object identifier is for all virtual drives on controller *x*.
/c*x*/e*x*	This object identifier is for an enclosure *x* on controller *x*.
/c*x*/eall	This object identifier is for all enclosures on controller *x*.
/c*x*/fall	This object identifier is for all foreign configurations on controller *x*.
/c*x*/ln*x*	This is the object identifier for the lane speed *x* on controller *x*
/c*x*/e*x*/s*x*	This object identifier for the drive is slot *x* on enclosure *x* on controller *x*.
/c*x*/s*x*	This object identifier represents the drives that are directly attached to controller *x*.
/call/sall	This object identifier represents all the drives that are directly attached to all controllers.
/c*x*/e*x*/sall	This object identifier is for all the drives on enclosure *x* on controller *x*.
/c*x*/d*x*	This object identifier is for the drive group *x* on enclosure *x* on controller *x*.
/c*x*/dall	This object identifier is for the all drive groups on enclosure *x* on controller *x*.
/c*x*/p*x*	This object identifier is for a phy operation *x* on controller *x*.
/c*x*/pall	This object identifier is for all phy operations on controller *x*.
/c*x*/bbu	This object identifier is for a BBU on controller *x*.
/c*x*/cv	This object identifier is for a cache vault *x* on controller *x*.
/c*x*/mx	This object identifier is a MUX identifier, cache vault *x* on controller *x*.

NOTE
If enclosures are not used to connect physical drives to the controller, you do not specify the enclosure ID in the command.

The StorCLI tool supports the following verbs.

Table 3: Verbs in the StorCLI Command Syntax

Verb	Description
add	This verb adds virtual drives, JBODs, and so on to the object identifier.
compare	This verb compares an input value with a system value.
del	This verb deletes a drive, value, or property of the object identifier.
download	This verb downloads and flashes a file to the target.
expand	This verb expands the size of the virtual drive.
erase	This verb erases a particular region on the controller, depending on the argument specified.
flush	This verb flushes a controller cache or a drive cache.
flasherase	This verb erases the flash memory on the controller.
get	This verb obtains the data from the controller.
import	This verb imports the foreign configuration into the drive.
insert	This verb replaces the configured drive that is identified as missing, and starts an automatic rebuild.
pause	This verb pauses an ongoing operation.
resume	This verb resumes paused operation.

Verb	Description
`restart`	This verb restarts the controller without a system reboot.
`set`	This verb sets a value of the object identifier.
`show`	This verb shows the value and properties of the object identifier.
`split`	This verb enables you to perform a break mirror operation on a drive group.
`suspend`	This verb suspends a particular operation that is being performed.
`start`	This verb starts an operation.
`stop`	This verb stops an operation that is in progress. A stopped process cannot be resumed.
`spinup`	This verb spins up the drives connected to the controller.
`spindown`	This verb spins down an unconfigured drive and prepares it for removal.
`shutdown`	This verb shuts down the controller.
`transform`	This verb downgrades the firmware memory on the controller.

- `<[adverb | attributes | properties]>`
  Specifies what the verb modifies or displays.
- `<[key=value]>`
  Specifies a value, if a value is required by the command.

StorCLI Default Logging

Default logging functionality has been enabled in StorCLI. When a default log file is created, the file is saved as `storcli.log` . Each time default logging occurs, the information is added to the `storcli.log` . Once the log file reaches a maximum size of 3 MB, a new log file is created. There can be up to four log files at any given time. For example:

- `storcli.log`
- `storcli.log.1`
- `storcli.log.2`
- `storcli.log.3`

Due to default logging, there is a space limitation in light operating systems such as VMware or UEFI.

> **NOTE**
>
> StorCLI default logging requires a minimum of 20 MB of free space.

There are two conditions under which StorCLI logging occurs.

- When the `storcliconf.ini` file is present in the same directory as the StorCLI binary.
  Logging happens to the file name specified in the `ini` file. This is useful in situations where default logging will not work.
  For example, a segmentation fault occurs or a crash happens in StorCLI binary. In these situations, collect a StorCLI log file by placing the `storcliconf.ini` file in the same working directory as StorCLI.
- When the `storcliconf.ini` file is not present in the same directory as the StorCLI binary.
  Default logging occurs automatically.

Use the `nolog` option to disable logging for any command.

For example, include the `nolog` option in the `storcli /cx show nolog` command to prevent default logging.

StorCLI Controller Commands

StorCLI is a command line utility tool. StorCLI is not case sensitive. The order in which you specify the command options should be the same as in this document in order to ensure proper command execution. Incorrect or duplicate values for variables could result in the last variable being executed or in a command failure.

The version of the StorCLI and the operating system on which StorCLI is being executed are also displayed at the beginning of StorCLI output.

This section describes the commands supported by StorCLI.

System Commands

System Show Commands

StorCLI supports the following system `show` commands:

```
storcli show
storcli show all
storcli show ctrlcount
storcli show help
storcli v
```

The detailed description for each command follows.

storcli show

This command shows a summary of controller and controller-associated information for the system. The summary includes the number of controllers, the host name, the operating system information, and the overview of existing configuration.

storcli show all

This command shows the list of controllers and controller-associated information, information about the drives that need attention, and advanced software options.

storcli show ctrlcount

This command shows the number of controllers connected to the system.

storcli show help

This command shows help for all commands at the system level.

storcli v

This command shows the version of the StorCLI. The version of the StorCLI and the operating system on which StorCLI is being executed are also displayed at the beginning of StorCLI output.

Controller Commands

Controller commands provide information and perform actions related to a specified controller. The StorCLI utility supports the controller commands described in this section.

Show and Set Controller Properties Commands

Table 4: Controller Commands Quick Reference Table

Commands	Value Range	Description
show `<properties>`	See Table 5.	Displays specific controller properties.
set `<properties>`	See Table 5.	Sets controller properties.
show	`all` : Shows all properties of the virtual drive. `freespace` : Shows the free space available in the controller. See Controller Show Commands.	Displays physical drive information.

This section provides command information for the `show` and `set` controller properties.

> **NOTE**
> You cannot set multiple properties with a single command.

storcli /*cx* show *<property>*

This command shows the current value of the specified property on the specified controller.

General example output:

```
storcli /c0 show bgirate
Controller = 0
Status = Success
Description = None
Controller Properties:
====================
----------------
Ctrl_Prop Value
----------------
BGI Rate  45%
----------------
```

You can show the following properties using the `storcli /cx show <property>` command.

```
storcli /cx show abortcconerror
storcli /cx show activityforlocate
storcli /cx show alarm
storcli /cx show aso
storcli /cx show autobgi
storcli /cx show autoconfig
storcli /cx show autorebuild
storcli /cx show badblocks
storcli /cx show backplane
storcli /cx show batterywarning
storcli /cx show bgi
storcli /cx show bios
storcli /cx show bootdrive
storcli /cx show bootwithpinnedcache
storcli /cx show cachebypass
storcli /cx show cacheflushint
```

```
storcli /cx show cc
storcli /cx show ccrate
storcli /cx show coercion
storcli /cx show configureautobalance
storcli /cx show consistencycheck|cc
storcli /cx show copyback
storcli /cx show directpdmapping
storcli /cx show dimmerswitch|ds
storcli /cx show dpm
storcli /cx show ds
storcli /cx show eccbucketleakrate
storcli /cx show eccbucketsize
storcli /cx show eghs
storcli /cx show erase
storcli /cx show failpdonsmarterror
storcli /cx show flush|flushcache cachecade
storcli /cx show flushwriteverify
storcli /cx show foreignautoimport
storcli /cx show hddthermalpollinterval
storcli /cx show init
storcli /cx show jbod
storcli /cx show largeiosupport
storcli /cx show loadbalancemode
storcli /cx show limitmaxratesata
storcli /cx show maintainpdfailhistory
storcli /cx show migrate
storcli /cx show migraterate
storcli /cx show ncq
storcli /cx show ocr
storcli /cx show parityreadcachebypass
storcli /cx show patrolread|pr
storcli /cx show pci
storcli /cx show perfmode
storcli /cx show personality
storcli /cx show pi
storcli /cx show prcorrectunconfiguredareas
storcli /cx show preservedcache
storcli /cx show profile
storcli /cx show prrate
storcli /cx show rebuildrate
storcli /cx show rehostinfo
storcli /cx show restorehotspare
storcli /cx show safeid
storcli /cx show smartpollinterval
storcli /cx show sgpioforce
storcli /cx show spinupdelay
storcli /cx show spinupdrivecount
storcli /cx show ssdthermalpollinterval
storcli /cx show time
storcli /cx show usefdeonlyencrypt
storcli /cx show wbsupport
```

storcli /*cx* set *<property>* = *<value>*

General example output:

```
storcli /c0 set bgirate=40
Controller = 0
Status = Success
Description = None
Controller Properties :
====================
---------------
Ctrl_Prop Value
---------------
BGI Rate  40%
---------------
```

The following commands are examples of the properties that can be set using the `storcli /cx set` *<property>=<value>* command structure.

> **NOTE**
> In the following list and table, setting a property to `on` enables that feature and setting a property to `off` disables that feature.

```
storcli /cx set abortcconerror=[on|off]
storcli /cx set activityforlocate=[on|off]
storcli /cx set alarm=[on|off|silence]
storcli /cx set autoconfig=[none|R0 [immediate]|JBOD] > [usecurrent]
storcli /cx set autoconfig [sesmgmt=[on|off]] [securesed=[on|off]] [multipath=[on|off]]
   [multiinit=[on|off]] [discardpinnedcache=<Val>] [failPDOnReadME=[on|off]]
   [Lowlatency=low|off]]
storcli /cx set backplane [mode=[0|1|2|3]] [expose=[on|off]]
storcli /cx set batterywarning=[on|off]
storcli /cx set bgirate=<value>
storcli /cx set bootwithpinnedcache=[on|off]
storcli /cx set cachebypass=[on|off]
storcli /cx set cacheflushinterval=<value>
storcli /cx set ccrate=<value>
storcli /cx set coercion=<value>
storcli /cx set consistencycheck|cc=[off|seq|conc] [delay=<value>]
   [starttime=<yyyy/mm/dd hh>] [excludevd=x-y,z]
storcli /cx set copyback=[on|off] type=[smartssd|smarthdd|all]
storcli /cx set dimmerswitch|ds=[on|off] [type=[1|2|4]]
storcli /cx set directpdmapping=[on|off]
storcli /cx set driveactivityled=[on|off]
storcli /cx set eccbucketleakrate=<value>
storcli /cx set eccbucketsize=<value>
storcli /cx set eghs [state=[on|off]] [smarter=[on|off]] [eug=[on|off]]
storcli /cx set failpdonsmarterror=[on|off]
storcli /cx set flushwriteverify=[on|off]
storcli /cx set foreignautoimport=[on|off]
storcli /cx set immediateio=[on|off]
storcli /cx set jbod=[on|off]
storcli /cx set loadbalancemode=<value>
storcli /cx set maintainpdfailhistory=[on|off]
storcli /cx set migraterate=<value>
```

```
storcli /cx set ncq=[on|off]
storcli /cx set parityreadcachebypass=<on|off>
storcli /cx set patrolread|pr[[=on mode=[auto|manual]]|[off]]
storcli /cx set perfmode=<value>
storcli /cx set personality=[RAID|HBA|JBOD]
storcli /cx set pi [state=[on|off]] [import=[on|off]]
storcli /cx set prcorrectunconfiguredareas=[on|off]
storcli /cx set profile profileid=<value>
storcli /cx set prrate=<value>
storcli /cx set rebuildrate=<value>
storcli /cx set restorehotspare=[on|off]
storcli /cx set sesmonitoring=[on|off]
storcli /cx set SGPIOforce=[on|off]
storcli /cx set smartpollinterval=<value>
storcli /cx set spinupdelay=<value>
storcli /cx set spinupdrivecount=<value>
storcli /cx set stoponerror=[on|off]
storcli /cx set supportssdpatrolread=[on|off]
storcli /cx set termlog[=on|off|offthisboot]
storcli /cx set time=<yyyymmdd hh:mm:ss|systemtime>
storcli /cx set usefdeonlyencrypt=[on|off]
```

The following table lists and describes the properties for the `show` and `set` commands.

Table 5: Properties for Show and Set Commands

Cmd	Property Name	Set Command Range	Description
set	abortcconerror	=[on|off]	Aborts the consistency check when it detects an inconsistency.
set	activityforlocate	=[on|off]	If set to on , the LEDs usually used to show drive activity are used for drive locate instead.
set	alarm	=[on|off|silence]	Enables (on) or disables (off) the alarm on critical errors. The option silence silences the alarm. This option is supported only on MegaRAID controllers that have a physical buzzer. If set to on , the alarm sounds on critical errors. If set to off , any active alarms are silenced and the alarm will not sound on any errors, including critical errors. If set to silence , any currently sounding alarm stops, but starts again after a reboot if another critical error is present and the alarm is on .
show	aso		Displays the aso status.
set	aso	deactivatetrialkey	Displays the enabled Advanced Software Options.
set	aso	key=<key value>	
set	aso	key=<key value> preview	

Cmd	Property Name	Set Command Range	Description
set	aso	rehostcomplete	
set	aso	transfertovault	
set	assemblynumber	=<*xxxx*>	—
show	autobgi		Displays the autobgi status.
show	autoconfig		Displays the autoconfig status.
set	autoconfig	=[R0|JBOD|NONE]	Sets the behavior to R0 , JBOD , or NONE .
show	autorebuild		Displays the autorebuild status.
set	autorebuild	=[on|off]	—
show	badblocks		Displays bad blocks of the CacheVault ™ module.
set	backplane	expose=[on|off]	Enables (on) or disables (off) device drivers to expose enclosure devices; for example, expanders, SEPs.
set	backplane	mode=0 : Use autodetect logic of backplanes, such as SGPIO and I^{2}C SEP using GPIO pins. mode=1 : Disable autodetect SGPIO. mode=2 : Disable I^{2}C SEP autodetect. mode=3 : Disable both the autodetects.	Configures enclosure detection on a non-SES/expander backplane.
set	batterywarning	=[on|off]	Enables (on) or disables (off) battery warnings.
show	bgi		Displays the bgi (background initialization) status.
set	bgirate	=0 to 100	Sets the background initialization rate in percentage.
show	bios		Displays the bios status.

set	bios	[state=[on|off]]
		  [Mode=[SOE|PE|IE|SME]]
		  [abs=[on|off]]
		[DeviceExposure=<*value*>]
			—

show	bootdrive		Displays the bootdrive status.
set	cachebypass	=[on|off]	Enables (on) or disables (off) the cache bypass performance improvement feature.
set	cacheflushint	=0 to 255, Default value: 4	Sets the cache flush interval in seconds.
show	cc		Displays the cc (consistency check) status.
set	ccrate	=0 to 100	Sets the consistency check rate in percentage.
set	coercion	=0 : No coercion =1 : 128 MB =2 : 1 GB	Sets the drive capacity in coercion mode. The *12Gb/s MegaRAID* ® *Tri-Mode Software User Guide* (MR-TM-SW-UG109) contains more information about coercion.

Cmd	Property Name	Set Command Range	Description
set	config	file=*<fileName>*	Saves the controller and the controller properties to the specified file.
set	configautobalance	=[on|off]	—
show	configautobalance		Displays the configautobalance status.
set	consistencycheck	=[on|off]	See Consistency Check.
set	copyback	=[on|off] type=[smartssd|smarthdd| all]	Enables or disables copyback for drive types. The available choices are: smartssd – Copyback enabled for SSD drives. smarthdd – Copyback enabled for HDD drives. all – Copyback enabled for both SSD drives and HDD drives.
set	debug	type=*<value>* option=*<value>* [level=*<value in hex>*] reset all	—
set	dimmerswitch|ds	See Virtual Drive Power Settings Commands.	See Virtual Drive Power Settings Commands.
set	directpdmapping	=[on|off]	When using enclosures, set the directpdmapping property to off to disable it; otherwise, set it to on to enable it.
show	dpm		Displays the dpm status.
set	driveactivityled	=[on|off]	Activates or deactivates the Drive Activity LED.
show	ds	=OFF type=[1|2|3|4] =ON type=[1|2] [properties] =ON type=[3|4] DefaultLdType=*<val>* [properties] [properties]	Displays the Dimmer Switch status.
set	eccbucketleakrate	0 to 65535	Sets the leak rate of the single-bit bucket in minutes (one entry removed per leak-rate).
set	eccbucketsize	0 to 255	Sets the size of ECC single-bit-error bucket (logs event when full).
set	eghs	eug=[on|off]	Enables (on) or disables (off) the commissioning of Unconfigured Good drives as Emergency Hot Spare (EHSP) drives.
set	eghs	smarter=[on|off]	Enables (on) or disables (off) the commissioning of Emergency Hot Spare (EHSP) drives for Predictive Failure (PFA) events.
set	eghs	state=[on|off]	Enables (on) or disables (off) the commissioning of otherwise incompatible global hot spare drives as Emergency Hot Spare (EHSP) drives.

Cmd	Property Name	Set Command Range	Description
show	erase		Displays the `erase` status.
set	factory defaults		—
set	failpdonsmarterror	=[on|off]	Enables (`on`) or disables (`off`) the *Fail PD on SMARTer* property.
show	flush|flushcache	cachecade	Displays the CacheCade ™ `flush` status.
set	flushwriteverify	=[on|off]	Enables (`on`) or disables (`off`) the Write Verify feature. This feature verifies if the data was written correctly to the cache before flushing the controller cache.
set	flushwriteverify	=[on|off]	—
show	foreignautoimport		`On` – Displays the `foreignautoimport` status on startup. `Off` – Does *not* display the `foreignautoimport` status automatically on startup.
set	foreignautoimport	=[on|off]	Imports a foreign configuration automatically, at boot.
show	hddthermalpollinterval		Displays the `hddthremalpollinterval` status.
set	HDDThermalPollInterval	=*<value>*	—
set	immediateio	=[on|off]	Enables (`on`) or disables (`off`) immediate I/O transactions.
show	init		Displays the `init` status.
set	jbod	=[on|off]	Enables (`on`) or disables (`off`) JBOD mode; by default, drives become system drives. This property is not supported by all controllers. **Note:** If you try to disable the JBOD mode, and if any of the JBODs have an operating system/file system, the StorCLI tool displays a warning message indicating that the JBOD has an operating system or a file system on it and prompts you to use the `force` option to proceed with the disable operation.
set	jbodwritecache	=[on|off|default]	—
show	jbodwritecache		Displays the `jbodwritecache` status.
set	largeiosupport	=[on|off]	—
show	largeiosupport		Displays the `largeiosupport` status.
set	largeQD	=[on|off]	—
show	largeQD		Displays the `largeQD` status.
set	ldlimit	=[default|max]	—
show	ldlimit		Displays the `ldlimit` status.
set	limitMaxRateSATA	=[on|off]	—
show	limitmaxratesata		Displays the `limitmaxratesata` status.

Cmd	Property Name	Set Command Range	Description
set	linkconfig	`[connname=c`*x*`,c`*y*`]` `linkconfig=<val>`	—
set	loadbalancemode	`=[on|off]`	Enables (`on`) or disables (`off`) automatic load balancing between SAS phys or ports in a wide port configuration.
show	maintainpdfailhistory		Displays the `maintainpdfailhistory` status.
set	maintainpdfailhistory	`=[on|off]`	`On` (enabled) – The controller keeps track of drives that have previously failed on the controller. If a previously failed drive is reattached to the controller, it is displayed as a foreign configuration and will need to be imported before it can be used. `Off` (disabled) – The controller does *not* track failed drives.
set	maintenance	`mode=[normal|nodevices]`	—
show	migrate		Displays the `migrate` status.
set	migraterate	`=0 to 100`	Sets the VD configuration migration rate in percentage.
show	ocr		Displays the online controller reset (`ocr`) status.
set	ocr	`=[on|off]`	—
show	parityreadcachebypass		Displays the `parityreadcachebypass` .
set	parityreadcachebypass	`=[on|off]`	Sets the `parityreadcachebypass` .
set	patrolread	`[starttime=<`*yyyy/mm/dd hh*`>] pr`	—
set	patrolread	`=[[on mode=[auto| manual]]|[off]] pr`	—
set	patrolread	`delay = <value>`	—
set	patrolread|pr	See Patrol Read.	See Patrol Read.
show	pci		Displays the `pci` status.

set	pdfaileventoptions	`[detectionType=<val>]`
		`[correctiveaction=<val>]`
		`[errorThreshold=<val>]`
			—

set	perfmode	`=<value>` `[maxflushlines=<value> numiostoorder=<value>]`	—
set	perfmode	`=0` : Tuned to provide the best IOPs, currently applicable to non-FastPath. `=1` : Tuned to provide the least latency, currently applicable to non-FastPath.	Performance tuning setting for the controller.
show	personality		Displays the current, supported, and requested personalities. It also displays the current behavior and respective behavior parameters.

Cmd	Property Name	Set Command Range	Description
set	personality	=[RAID|HBA|JBOD]	Sets the personality to RAID , JBOD , or HBA . If you switch personalities, you must reboot the system for the changes to take effect.
set	pi	=[on|off]	Enables (on) or disables (off) Protection Information (sometimes called data protection) support on the controller.
set	pi	import=[on|off]	Enables (on) or disables (off) import data protection drives on the controller.
set	prcorrectunconfiguredareas	=[on|off]	Corrects media errors during patrol read by writing 0s to unconfigured areas of the disk.
show	preservedcache		Displays the preservedcache status.
set	profile	profileid=<id>	Sets the profile ID. Valid profile values (set using profileid=option) differ for each MR controller model and firmware version; consult the applicable MegaRAID controller user guide and release notes document for supported profileid values.
set	prrate	=0 to 100	Sets the patrol read rate of the virtual drives in percentage.
set	rebuildrate	=0 to 100	Sets the rebuild rate of the drive in percentage.
set	reconrate	=0 to 100	Sets the reconstruction rate for a virtual drive, as a percentage.
set	restorehotspare	=[on|off]	Becomes a hot spare on insertion of a failed drive.
set	sasadd	= <xxxx> [devicename] [methodport]	—
set	sasaddhi	= <xxxx> [devicename] [methodport]	—
set	sasaddlow	= xxxxx [devicename] [methodport]	—
set	securitykey	=<xxxxxxxx> [passphrase=<xxxx>] [keyid=<xxx>]	
set	securitykey	keyid=<xxx>	—
set	sesmonitoring	=[on|off]	Enables (on) or disables (off) SES monitoring.
show	sgpioforce		Displays the sgpioforce status.
set	SGPIOforce	=[on|off]	Forces the SGPIO status per port only for four drives; affects high performance computing (HPC) controllers.

Cmd	Property Name	Set Command Range	Description
`set`	`smartpollinterval`	=0 to 65535	Sets the time for polling of SMART errors, in seconds.
`show`	`spinupdelay`		Displays the `spinupdelay` status.
`set`	`spinupdelay`	=0 to 255	Sets the spin-up delay between a group of drives or a set of drives, in seconds.
`set`	`spinupdrivecount`	=0 to 255	Sets the number of drives that are spun up at a time.
`show`	`ssdthermalpollinterval`		Displays the `ssdthermalpollinterval` status.
`set`	`SSDThermalPollInterval`	=`<value>`	—
`set`	`stoponerror`	=[on|off]	Stops the MegaRAID BIOS during POST, if any errors are encountered.
`set`	`supportssdpatrolread`	=[on|off]	Enables (`on`) or disables (`off`) patrol read for SSD drives.
`set`	`supportssdpatrolread`	=[on|off] pr	—
`set`	`termlog`	=[on|off|offthisboot]	Enables (`on`) or disables (`off`) the `termlog` to be flushed from DDR to ONFI (Open NAND Flash Interface). `offthisboot` – Disables the `termlog` flushes to ONFI only for this boot. In the next boot, the `termlog` is enabled.
`set`	`time`	Valid time in `<yymmdd hh:mm:ss>` format or `systemtime`	Sets the controller time to your input value or the system time (local time in 24-hour format).
`set`	`tracernumber`	= `<xxxx>`	—
`set`	`updatevpd`	file= `<filepath>`	—
`set`	`usefdeonlyencrypt`	=[on|off]	Enables (`on`) or disables (`off`) FDE drive-based encryption.

Controller Show Commands

StorCLI supports the following `show` commands:

```
storcli /cx show
storcli /cx show all [logfile[=filename]]
```

The detailed description for each command follows.

storcli /cx show

This command shows the summary of the controller information. The summary includes basic controller information, foreign configurations, drive groups, virtual drives, physical drives, enclosures, and BBU information.

Input example:

```
storcli /c1 show
```

storcli /cx show all [logfile[=*filename*]]

The `cx show all` command shows all of the controller information, which includes basic controller information, bus information, controller status, advanced software options, controller policies, controller defaults, controller capabilities, scheduled tasks, miscellaneous properties, foreign configurations, drive groups, virtual drives, physical drives, enclosures, and BBU information.

If you use the `logfile` option in the command syntax, the logs are written to the specified file. If you do not specify the file name, then the logs are written to the `storsas.log` file. If you do not use the `logfile` option in the command syntax, the entire log output is printed to the console.

Ensure that the filename does not contain a blank space.

Input examples:

```
storcli /c0 show all logfile=log.txt
storcli /c0 show all logfile = abc.txt
```

> **NOTE**
> The PCI information displayed as part of the `storcli /cx show` and `storcli /cx show all` commands is not applicable for the FreeBSD operating system. Hence, the PCI information fields are displayed as `N/A`.

Controller Debug Commands

The Storage Command Line Tool supports the following debug commands. There should be at least 20MB of free space for StorCLI to perform debug logging.

Syntax

```
storcli /cx set debug type = <value> option = <value> level = [<value in hex>]
```

This command enables the firmware debug variables.

Where:

- `/cx` – Specifies the controller where x is the index of the controller.
- `type` – Takes the value from 0 to 128, mapping each number to a particular debug variable in the firmware.
- `option` – Takes the value from 0 to 4, where:
  - 0: NA
  - 1: SET
  - 2: CLEAR
  - 3: CLEAR ALL
  - 4: DEBUG DUMP
- `level` – Supports multiple levels of debugging in the firmware.

Syntax

```
storcli /cx set debug reset all
```

This command enables the firmware debug logs from the application.

Where:

`/cx` – specifies the controller where x is the index of the controller.

> **NOTE**
> The `debug type`, the `debug value`, and the `debug level` parameters for the preceding `debug` commands are exclusively used by the Broadcom Technical Support Team to provide technical support. For assistance with these debug commands, contact Broadcom Technical Support representative.

Controller Background Task Operation Commands

Profile Management

On controllers that support profile management, the StorCLI utility supports the following profile management commands:

```
storcli /cx show profile
```

```
storcli /cx set profile profileid=<value>
```

The detailed description for each command follows:

storcli /cx show profile

This command displays the profiles supported by the controller.

> **NOTE**
> Only Broadcom Tri-Mode MegaRAID controllers support profiles.

Input example:

```
storcli /c0 show profile
```

Output example:

On successful execution of the command, the output will have the following fields:

- `Mode`
  The mode supported by the current controller profile (HBA, RAID, JBOD).
- `ProfileID`
  Displays the current profile ID.
- `MaxPhyDrv`
  Displays the maximum number of physical drives supported.
- `MaxLD`
  Displays the maximum number of logical drives supported.
- `MaxNVMeDev`
  Displays the maximum number of NVMe drives supported.
- `MaxAHCIDev`
  Displays the maximum number of AHCI devices supported.
- `isDefault`
  Displays if the displayed profile ID is the same as the default profile ID.
- `isCurrent`
  Displays if the displayed profile ID is the same as the current profile ID

storcli /cx set profile profileid= <value>

This command sets the specified profile ID of the controller. You need to specify the profile ID in decimal format. For the Profile ID to change, a system reboot is required.

Input example:

```
storcli /c0 set profile profileid=11
```

> **NOTE**
> The maximum number and type of PDs supported depends on the profile ID that is selected. Check the applicable MegaRAID user guide and firmware release notes for these values.

Rebuild Rate

```
storcli /cx set rebuildrate=<value>
storcli /cx show rebuildrate
```

The detailed description for each command follows.

storcli /*cx* set rebuildrate=<*value*>

This command sets the rebuild task rate of the specified controller. The input value is in percentage.

Input example:

```
storcli /c0 set rebuildrate=30
```

> **NOTE**
> A high rebuild rate slows down I/O transaction processing.

storcli /*cx* show rebuildrate

This command shows the current rebuild task rate of the specified controller in percentage.

Input example:

```
storcli /c1 show rebuildrate
```

Patrol Read

The Storage Command Line Interface Tool supports the following patrol read commands:

```
storcli /cx resume patrolread
storcli /cx set patrolread =[on [mode=[auto|manual]]]|[off]
storcli /cx set patrolread [starttime=<yyyy/mm/dd hh>] [maxconcurrentpd=<value>] [includessds=[on|off]]
 [uncfgareas=[on|off]]
storcli /cx set patrolread  delay=<value>
storcli /cx show patrolread
storcli /cx start patrolread
storcli /cx stop patrolread
storcli /cx pause patrolread
```

> **NOTE**
> A patrol read operation is scheduled for all the online physical drives of the controller.

The detailed description for each command follows.

storcli /*cx* resume patrolread

This command resumes a suspended patrol read operation.

Input example:

```
storcli /c0 resume patrolread
```

storcli /*cx* set patrolread=[on [mode=[auto|manual]]]|[off]

This command turns the patrol read scheduling on and sets the mode of the patrol read to automatic or manual.

Input example:

```
storcli /c0 set patrolread=on mode=manual
```

storcli /*cx* set patrolread [starttime=<*yyyy/mm/dd hh*>] [maxconcurrentpd=<*value*>] [includessds=[on|off]] [uncfgareas=o[n|off]]

This command schedules a patrol read operation. You can use the following options for patrol read command operations.

Table 6: Set Patrol Read Input Options

Option	Value Range	Description
starttime	A valid date and hour in 24-hour format	Sets the start time in *yyyy/mm/dd hh* format.
maxconcurrentpd	Valid number of physical drives present	Sets the number of physical drives that can perform patrol read at a single time.
includessds	on or off	Includes SSDs in the patrol read operation.
uncfgareas	on or off	Includes the areas not configured in the patrol read process.

> **NOTE**
> Controller time is taken as a reference for scheduling a patrol read operation.

Input example:

```
storcli /c0 set patrolread=on starttime=2012/02/21 00
```

storcli /*cx* set patrolread [delay=<*value*>]

This command delays the scheduled patrol read in hours.

Input example:

```
storcli /c0 set patrolread delay=30
```

storcli /*cx* show patrolRead

This command shows the current state of the patrol read operation along with other details, such as the **PR Mode**, **PR Execution Delay**, **PR iterations completed**, and **PR on SSD**. This command also shows the start time and the date when the patrol read operation started.

The values shown for the current state of the patrol read operation are **Ready**, **Active**, **Paused**, **Aborted**, **Stopped**, or **Unknown**.

If the state of the patrol read is active, a numeric value is shown along with the state that depicts the number of physical drives that have completed the patrol read operation. As an example, `Active 1` means that the one physical drive has completed the patrol read operation.

Input example:

```
storcli /c0 show patrolread
```

storcli /*cx* start patrolread

This command starts the patrol read operation. This command starts a patrol read operation immediately.

Input example:

```
storcli /c0 start patrolread
```

storcli /*cx* stop patrolread

This command stops a running patrol read operation.

Input example:

```
storcli /c0 stop patrolread
```

> **NOTE**
> You cannot resume a stopped patrol read operation.

storcli /*cx* pause patrolread

This command pauses a running patrol read operation.

Input example:

```
storcli /c0 pause patrolread
```

> **NOTE**
> You can run this command only when a patrol read operation is running on the controller.

Consistency Check

The StorCLI utility supports the following commands to schedule, perform, and view the status of a consistency check (CC) operation:

```
storcli /cx set consistencycheck|cc=[off|seq|conc][delay=<value>] starttime=<yyyy/mm/dd hh> [excludevd=x-y,z]
storcli /cx show cc
storcli /cx show ccrate
```

The detailed description for each command follows.

storcli /*cx* set consistencycheck|cc=[off|seq|conc][delay=<*value*>] starttime=<*yyyy/mm/dd hh*> [excludevd=*x-y,z*]

This command schedules a consistency check (CC) operation. You can use the following options with the consistency check command.

Table 7: Set CC Input Options

Option	Value Range	Description
cc	seq – Sequential mode. conc – Concurrent mode. off – Turns off the consistency check.	Sets CC to either sequential mode or concurrent mode, or turns off the CC. The concurrent mode slows I/O processing.
delay	−1 and any integer value.	Delays a scheduled consistency check. The value is in hours. A value of 0 makes the CC runs continuously with no delay (in a loop). Only scheduled consistency checks can be delayed.
starttime	A valid date and hour in 24-hour format.	The start time of a consistency check is *yyyy/mm/dd hh* format.

Option	Value Range	Description
excludevd	The range should be less than the number of virtual drives.	Excludes virtual drives from the consistency checks. To exclude particular virtual drives, you can provide list of virtual drive names (V*x*,V*y* … format) or the range of virtual drives that you want to exclude from a consistency check (V*x*-V*y* format). If this option is not specified in the command, no virtual drives are excluded.

Input example:

```
storcli /c0 set CC=on starttime=2012/02/21 00 excludevd=v0-v3
```

storcli /*cx* show cc

This command shows the consistency check schedule properties for a controller.

Input example:

```
storcli /c0 show cc
```

storcli /*cx* show ccrate

This command checks the status of a consistency check operation. The CC rate appears in percentage.

Input example:

```
storcli /c0 show ccrate
```

> **NOTE**
> A high CC rate slows I/O processing.

Premium Feature Key Commands

The Storage Command Line Interface Tool supports the following commands for premium feature keys:

```
storcli /cx set advancedsoftwareoptions(aso) key=<value> [preview]
storcli /cx show aso
storcli /cx set aso [transfertovault][rehostcomplete][deactivatetrialkey]
storcli /cx show safeid
```

The detailed description for the command follows.

storcli /*cx* set advancedsoftwareoptions(aso) key=*<value>* [preview]

This command activates advanced software options (ASO) for a controller. You can use the following options with the advanced software options command.

Table 8: Set Advanced Software Options Input Options

Option	Value Range	Description
key	40 alphanumeric characters.	The key to activate the ASO on the controller. After they are activated, ASOs cannot be removed from the controller.
deactivatetrialkey	—	Deactivates any currently active trial key.

Option	Value Range	Description
`rehostcomplete`	—	Enables rehosting on the specified controller.
`transfertovault`	—	Transfers the ASO key to the vault and disables the ASO.

Input example:

```
storcli /c0 set aso key=LSI0000
```

storcli /*cx* show safeid

This command shows the Safe ID of the specified controller.

Input example:

```
storcli /c0 show safeid
```

Controller Security Commands

The StorCLI Utility supports the following controller security commands:

```
storcli /cx compare securitykey < =xxxxxxxx | file=filename >
storcli /cx delete securitykey
storcli /cx set securitykey < keyid=xxx | file=filename >
storcli /cx set securitykey < =xxxxxxxx  {passphrase=xxxx} {keyid=xxx} | file=filename >
storcli /cx set securitykey < =xxxxxxxx oldsecuritykey=xxxxxxxx {passphrase=xxxx} [keyid=xxx] | file=filename>
storcli /cx [/ex]/sx set security=on
```

The detailed description for each command follows.

storcli /*cx* show securitykey keyid

This command shows the security key on the controller.

Input example:

```
storcli /c0 show securityKey keyid
```

storcli /*cx* compare securitykey < =*xxxxxxxx* | file=*filename* >

This command compares and verifies the security key of the controller.

storcli /*cx* delete securitykey

This command deletes the security key of the controller.

Input example:

```
storcli /c0 delete securitykey
```

storcli /*cx* set securitykey < keyid=*xxx* | file=*filename* >

This command sets the key ID for the controller. The key ID is unique for every controller.

storcli /c*x* set securitykey < =*xxxxxxxx* {passphrase=*xxxx*} {keyid=*xxx*} | file=*filename* >]

This command sets the security key for the controller. You can use the following options with the `set security key` command.

Table 9: Set Security Key Input Options

Option	Value Range	Description
`passphrase`	Should have a combination of numbers, uppercase letters, lowercase letters, and special characters. Minimum of 8 characters and maximum of 32 characters.	A string that is linked to the controller and is used in the next bootup to encrypt the lock key. If `passphrase` is not set, the controller generates it by default.
`keyid`	—	The unique ID set for different controllers to help you specify a passphrase to a specific controller.

Input example:

```
storcli /c0 set securitykey=Lsi@12345 passphrase=Lsi@123456 keyid=1
```

storcli /c*x* set securitykey < =*xxxxxxxx* oldsecuritykey=*xxxxxxxx* {passphrase=*xxxx*} [keyid=*xxx*] | file=*filename*>

This command changes the security key for the controller.

Input example:

```
storcli /c0 set securitykey=Lsi@12345 oldsecuritykey=pass123 passphrase=Lsi@123456 keyid=1
```

storcli /c*x*[/e*x*]/s*x* set security=on

This command sets the security on the FD-capable JBOD drive.

Input example:

```
storcli /c0/e0/s0 set security=on
```

Flashing Controller Firmware Command while the Firmware Is Operational

> **NOTE**
> The Flashing Controller Firmware Command while the Firmware Is Operational is not supported in Embedded MegaRAID.

storcli /c*x* download file=<*filepath*> [fwtype=<*value*>] [nosigchk] [noverchk] [resetnow] [force] [forceclose]

This command flashes the firmware with the ROM file to the specified adapter from the given file location (`<filepath>` is the absolute file path).

You can use the following options in the table to flash the firmware.

Table 10: Flashing Controller Firmware Input Options

Option	Value Range	Description
`file`	`filepath`	The absolute file path.
`nosigchk`	—	The application flashes the firmware even if the check word on the file does not match the required check word for the controller. You can damage the controller if a corrupted image is flashed using this option.
`noverchk`	—	The application flashes the controller firmware without checking the version of the firmware image. This option must be provided between phases to the downgrade firmware.
`fwtype`	0 : Application 1 : TMMC 2 : GG-Enhanced	The firmware type to be downloaded. The application downloads the firmware for the controller. The TMMC downloads the firmware for the TMMC battery only. Default is 0 (application).
`resetnow`	—	Invokes online firmware update on the controller; you do not need to reboot the controller to make the update effective. The `resetnow` option is not supported in the UEFI mode.
`[forceclose]`	—	Used for calling the flash close before start of the FW download process.

Input example:

```
storcli /c1 download file=c:\app.rom fwtype=0
```

Flashing Controller Firmware Command while the Firmware Is Nonoperational

> **NOTE**
> This command is only valid for flashing SAS HBAs, not MegaRAID controllers. This command should be run only under the direction of FAE or Broadcom support. The support team will provide the files and guidance required.

storcli /cx download completeflash fileone=<_IT boot loader image_> filetwo=<_firmware image_>

This command downloads the complete flash image on a non-operational or an empty controller by performing host boot. This command takes two files as arguments:

- `fileone` – A valid `Itboot` loader image with which host boot is performed on the controller.
- `filetwo` – A valid firmware package that is flashed on the controller.

Input example:

```
storcli /c1 download completeflash fileone=<Itbootloaderimage> filetwo=<FW image>
```

Erase Command

storcli /cx erase all [excludemfg]

This command erases the complete controller flash region but retains the manufacturing data region.

Input example:

```
storcli /c0 erase all excludemfg
```

Controller Cache Command

The following command flushes the controller cache.

storcli /c*x* flush|flushcache

This command flushes the controller cache.

Input example:

```
storcli /c0 flushcache
```

Controller Configuration Commands

The following commands work with the controller configuration.

storcli /c*x* set config file=*<file_name>*

This command obtains the controller configuration and its properties from the specified file.

> **NOTE**
>
> You cannot load a saved configuration over an existing configuration when there are existing virtual drives. To load a saved configuration, you must first clear the existing configuration on the target controller.
>
> Save and restore functionality is intended to restore the saved configuration within the scope of the controller. Preserving the operating system disk labels is beyond the scope of the save and restore feature. On previous generation products, the selection of target IDs by firmware may result in the operating system restoring the same disk labels for RAID volumes on certain systems, but with SAS38xx and SAS39xx controllers target ID range allocated for RAID volumes does not ensure any disk label ordering at the operating system level.

Input example:

```
storcli /c0 set config file=log.txt
```

storcli /c*x* get config file=*<file_name>*

This command saves the controller configuration and its properties to the specified file.

Input example:

```
storcli /c0 get config file=log.txt
```

SnapDump Commands

Snapshot dumping is a mechanism of saving a snapshot of the debug information at fault time. The intention is to collect all required information to be able to root-cause the defect at the first instance of defect detection. The SnapDump command makes sure that multiple defect reproductions are not required to debug.

Windows Driver RTTrace

Use the following commands to collect the RTTrace files:

```
storcli /c0 get snapdump id=<snapDumpId> file=<filename>.zip

storcli get rttdump

storcli /c0 set snapdump preboottrace=on/off
```

Detailed descriptions of each command follows.

storcli /*cx* get snapdump id=*<snapDumpId>* file=*<filename>*.zip

This is the SnapDump zip filename.

storcli get rttdump

This command retrieves the RTT file directly from the driver.

Input example:

```
storcli get rttdump
```

storcli /*cx* set snapdump preboottrace=on/off

This command switches the preboot trace on and off.

Input example:

```
storcli /c0 set snapdump preboottrace=on
```

SnapDump Support Commands

The StorCLI utility checks the SnapDump support bit and allows the user to `show` or `set` the controller on-off property for SnapDump. Using the support command, you can enable or disable the SnapDump feature on the controller using the commands that follow.

```
storcli /cx set snapdump state=on|off
```

```
storcli /cx show snapdump
```

Detailed descriptions of each command follows.

storcli /*cx* set snapdump state=on|off

This command enables or disables the SnapDump feature on the controller.

Input example:

```
storcli /c0 set snapdump state=on
```

storcli /*cx* show snapdump

The `show snapdump` command displays whether the SnapDump feature is enabled or not. If enabled, a detailed list of SnapDump properties is displayed, such as the number of dumps that firmware can save and delay for OCR. This command also displays the list of SnapDump files that the firmware currently has, the sizes of the files, and the time the SnapDump is generated, and whether it is an on-demand SnapDump or auto-generated one based on the firmware capability.

Input example:

```
storcli /c0 show snapdump
```

Modifying SnapDump Properties Command

With the SnapDump feature enabled, the user can set the various other properties of the SnapDump, such as the number of dumps that firmware can save and delay for OCR.

storcli /*cx* set snapdump [savecount=<*value*> | delayocr=<*value*>]

Where:

- `savecount` –
  - For the SAS3516 controller, `savecount` sets the number of times the SnapDump will persist in firmware, in case the user does not collect the data.
  - For all other controllers, `savecount` is no longer used and is displayed as `N/A`.
- `delayocr` – Delays the driver trigger for SnapDump before it initiates OCR; the delay is in seconds.

Input example:

```
storcli /c0 set snapdump [savecount=<value>] [delayocr=<value>]
```

Retrieving SnapDump Data Commands

The StorCLI utility supports the SnapDump commands that follow.

```
storcli /cx get snapdump ID=<val> file=<filename>
storcli /cx get snapdump ID=all
storcli /cx get snapdump
```

Detailed descriptions for each command follows.

storcli /*cx* get snapdump ID=<*val*> file=<*filename*>

To download a specific SnapDump ID, you must read the ID from the firmware. The StorCLI utility keeps writing the data to the file, truncating the file and adding new information.

Input example:

```
storcli /c0 get snapdump ID=<val> file=<filename>
```

Where:

- `val` – Specifies the SnapDump ID number.
- filename – Specifies the file, in zip format, in which to write the SnapDump data

storcli /*cx* get snapdump ID=all

To download all SnapDump IDs that are present on the controller, use the `all` option.
With this command, the file name is framed by the CLI in a specific format as shown:
`snapdump_c#(controllerid)_id#(snapdump_id)_year_month_day_hour_min_sec.zip`

Input example:

```
storcli /c0 get snapdump ID=all
```

storcli /*cx* get snapdump
To generate and download all SnapDump data when the user has not provided ID, an on-Demand
request to the controller is generated and downloads all the SnapDump data present on the
controller. With this command, the file name is framed by the CLI in a specific format as shown:
`.snapdump_c#(controllerid)_id#(snapdump_id)_year_month_day_hour_min_sec.zip`

Input example:

```
storcli /c0 get snapdump
```

Clearing SnapDump Data Commands

The StorCLI utility has an option to delete all the SnapDump data from the firmware.

> **NOTE**
> It is recommended that you save all previous snapdumps, as personality changes and flashing a new firmware
> package will discard all snapdumps on both DDR and flash.

```
storcli /cx delete snapdump [force]
```

A detailed description for this command follows.

storcli /cx delete snapdump [force]

To clear the SnapDump data from the firmware, use this command application to request the firmware to clear/delete the
SnapDump data. If the `force` option is not specified, the StorCLI tool warns the user that this command will clear the
SnapDump data and prompt the user to use the `force` option. When the `force` option is specified, the CLI requests the
firmware to clear all the SnapDump data.

Input example:
```
storcli /c0 delete snapdump [force]
```

SPDM Commands

StorCLI SPDM commands display the security protocol details and allow users to configure the security protocol on a
controller. The SPDM commands allow users to view the security protocol version, slot status, export and import security
protocol, and invalidate a slot.

```
storcli /cx show
storcli /cx show all
storcli /cx show security spdm slotgroup=xx slot=yy
storcli /cx export security spdm slotgroup=0 slot=1 subject=subjectname file=filename
storcli /cx import security spdm slotgroup=xx slot=yy file=filename [seal]
storcli /cx set security spdm slotgroup=xx slot=yy invalidate
storcli /cx get security spdm slotgroup=xx slot=yy file=filename
```

storcli /cx show

This command displays the security protocol support and security protocol properties detailed information.

storcli /cx show all

This command displays the security protocol information.

storcli /cx show security spdm slotgroup=xx slot=yy

This command reports the status of the certificate slot chain.

storcli /cx export security spdm slotgroup=0 slot=x1 subject=subjectname file=filename

This command requests the firmware create a certificate signing request and return it. The firmware returns an error if the
requested slot is already populated and sealed, if the slot group is invalid, or if the firmware cannot support the requested
BaseAsymAlgo and BaseHashAlgo fields selected.

storcli /c*x* import security spdm slotgroup=*xx* slot=*yy* file=*filename* [seal]

This command supplies a certificate chain from the application to the firmware. The firmware returns an error if the requested slot and slot group fields do not match an open session.

storcli /c*x* set security spdm slotgroup=*xx* slot=*yy* invalidate

This command invalidates the certificate chain storage slot.

storcli /c*x* get security spdm slotgroup=*xx* slot=*yy* file=*filename*

This command reads the certificate from the chain storage slot and allows the users to validate a downloaded certificate chain.

Diagnostic Command

The StorCLI Utility supports the following diagnostic command:

```
storcli /cx start diag duration=<val>
```

The detailed description for the command follows.

storcli /c*x* start diag duration=<*val*>

This command runs the diagnostic self-check on the controller for the specified time period in seconds.

Input example:

```
storcli /c0 start diag duration=5
```

> **NOTE**
> Ensure no IOs are running while executing this command.

Drive Commands

This section describes the drive commands, which provide information and perform actions related to physical drives. The following table describes frequently used drive commands.

Table 11: Physical Drives Commands Quick Reference Table

Commands	Value Range	Description
set	`missing` : Sets the drive status as missing. `good` : Sets the drive status to unconfigured good. `offline` : Sets the drive status to offline. `online` : Sets the drive status to online.	Sets physical drive properties.
show	`all` : Shows all properties of the physical drive. See Drive Show Commands.	Shows virtual drive information.

Drive Show Commands

The StorCLI utility supports the following drive `show` commands:

```
storcli /cx[/ex]/sx show
storcli /cx[/eall]/sall show
```

```
storcli /cx[/ex]/sx|sall show all
```

```
storcli /cx[/ex]/sx show smart
```

> **NOTE**
> If enclosures are used to connect physical drives to the controller, specify the enclosure ID in the command. If no enclosures are used, you must specify the controller ID and the slot ID.

The detailed description for each command follows.

storcli /c*x*[/e*x*]/s*x* show

This command shows the summary of the physical drive for a specified slot in the controller.

Input example:

```
storcli /c0/e25/s4 show
```

storcli /c*x*[/eall]/sall show

This command shows the summary information for all the enclosures and physical drives connected to the controller.

Input example:

```
storcli /c0/eall/sall show
```

storcli /c*x*[/e*x*]/s*x*|sall show all

This command shows all information of a physical drive for the specified slot in the controller. If you use the `all` option, the command shows information for all slots on the controller. *x* stands for a number, a list of numbers, a range of numbers, or all numbers.

This command also shows the NCQ (Native Command Queuing) status (**Enabled**, **Disabled**, or **N/A**), which is applicable only to SATA drives. If the controller to which the SATA drive is connected supports NCQ, and NCQ is enabled on the SATA drive, the status is shown as **Enabled**; otherwise, it is shown as **Disabled**. If NCQ is not a supported drive operation on the controller, the status is shown as **N/A**.

Input examples:

```
storcli /c0/e25/s0-3 show all
storcli /c0/e25/sall show all
```

> **NOTE**
> The `storcli /cx/sx show all` command shows drive information for the drive specified.

storcli /c*x*/[e*x*]/s*x* show smart

This command displays the SMART information of a SATA drive.

Input example:

```
storcli /c0/e25/s4 show smart
```

Missing Drives Commands

The StorCLI utility supports the following commands to mark and replace missing physical drives with the specified Unconfigured Good drive:

```
storcli /cx[/ex]/sx set missing
```

```
storcli /cx[/ex]/sx insert dg=A array=B row=C
storcli /cx[/ex]/sx set offline
storcli /cx/dall
```

The detailed description for each command follows.

> **NOTE**
> To set a drive that is part of an array as `missing`, first set it as `offline`. After the drive is set to `offline`, you can then set the drive to `missing`.

storcli /cx[/ex]/sx set missing

This command marks a drive as missing.

Input example:

```
storcli /c0/s4 set missing
```

storcli /cx[/ex]/sx insert dg=A array=B row=C

This command replaces the configured drive that is identified as missing, and then starts an automatic rebuild.

Input example:

```
storcli /c0/e25/s3 insert dg=0 array=2 row=1
```

storcli /cx[/ex]/sx set offline

This command marks the drive in an array as offline.

Input example:

```
storcli /c0/s4 set offline
```

storcli /cx/dall

This command finds the missing drives.

Set Drive State Commands

The StorCLI utility supports the following commands to set the status of physical drives:

```
storcli /cx[/ex]/sx set jbod
storcli /cx[/ex]/sx set good [force]
storcli /cx[/ex]/sx set offline
storcli /cx[/ex]/sx set online
storcli /cx[/ex]/sx set missing
storcli /cx[/ex]/sx set bootdrive=<on|off>
```

The detailed description for each command follows.

storcli /cx[/ex]/sx set jbod

This command sets the drive state to JBOD.

Input example:

```
storcli /c0/e25/s4 set jbod
```

storcli /c*x*[/e*x*]/s*x* set good [force]

This command changes the drive state to unconfigured good.

Input example:

```
storcli /c0/e25/s4 set good
```

> **NOTE**
>
> If the drive has an operating system or a file system on it, the StorCLI utility displays an error message and fails the conversion. If you want to proceed with the conversion, use the `force` option as shown in the following command.

Input example:

```
storcli /c0/e25/s4 set good [force]
```

storcli /c*x*[/e*x*]/s*x* set offline

This command changes the drive state to offline.

Input example:

```
storcli /c0/e25/s4 set offline
```

storcli /c*x*[/e*x*]/s*x* set online

This command changes the drive state to online.

Input example:

```
storcli /c0/e25/s4 set online
```

storcli /c*x*[/e*x*]/s*x* set missing

This command marks a drive as missing.

Input example:

```
storcli /c0/e25/s4 set missing
```

storcli /c*x*[/e*x*]/s*x* set bootdrive=<on|off>

This command sets or unsets a physical drive as a boot drive.

Input example:

```
storcli /c0/e25/s4 set bootdrive=on
```

Drive Initialization Commands

When you initialize drives, all the data from the drives is cleared. The StorCLI utility supports the following commands to initialize drives:

```
storcli /cx[/ex]/sx show initialization
storcli /cx[/ex]/sx start initialization
storcli /cx[/ex]/sx stop initialization
```

The detailed description for each command follows.

storcli /c*x*[/e*x*]/s*x* show initialization

This command shows the current progress of the initialization progress in percentage.

The estimated time (in minutes) left to complete the operation is also shown.

Input example:

```
storcli /c0/e25/s4 show initialization
```

storcli /c*x*[/e*x*]/s*x* start initialization

This command starts the initialization process on a drive.

Input example:

```
storcli /c0/e25/s4 start initialization
```

storcli /c*x*[/e*x*]/s*x* stop initialization

This command stops an initialization process running on the specified drive. A stopped initialization process cannot be resumed.

Input example:

```
storcli /c0/e25/s4 stop initialization
```

NVMe Drive Commands

The StorCLI Utility supports the following NVMe drive commands.

```
storcli/cx show failedNvmeDevices
storcli/cx[/ex]/sx show repair
storcli/cx[/ex]/sx start repair [force]
storcli/cx[/ex]/sx stop repair
```

Show Failed NVMe Devices

This command displays the list of initialization failed NVMe drives.

Syntax

```
storcli/cx show failedNvmeDevices
```

Show NVMe Drive Repair Status

This command displays the NVMe drive repair status.

Syntax

```
storcli/cx[/ex]/sx show repair
```

Start NVMe Drive Repair

This command starts the repair process on the requested drive.

Syntax

```
storcli/cx[/ex]/sx start repair [force]
```

Force – Deletes all data present on the drive.

Stop NVMe Drive Repair

This command stops the repair of the requested NVMe drive.

Syntax

```
storcli/cx[/ex]/sx stop repair
```

> **NOTE**
> If NVMe failed drives are detected, the controller state will move to `Need Attention`.

Drive Firmware Download Commands

The StorCLI Utility supports the following commands to download the drive firmware.

```
storcli /cx[/ex]/sx download src=filepath [satabridge] [mode= 5|7]

storcli /cx[/ex]/sx download src=<filepath> mode= E offline [activatenow [delay=<value>]]

storcli /cx[/ex]/sx download mode=F offline [delay=<value>]
```

storcli /cx[/ex]/sx download src=*filepath* [satabridge] [mode= 5|7]

This command flashes the drive firmware with the specified file.

The `satabridge` option lets you download the SATA bridge firmware in online mode.

The `mode` options specify the SCSI write buffer mode. The description follows:

- `5` – The entire drive firmware file is downloaded at once.
- `7` – The drive firmware file is downloaded in chunks of 32 KB.

> **NOTE**
> The default mode is `7` .

Input example:

```
storcli /c0/e56/s1 download src=c:\file1.bin

storcli /c0/e56/s1 download src=c:\file1.bin mode=5
```

storcli /cx[/ex]/sx download src=*<filepath>* mode= E offline [activatenow [delay=*<value>*]]

storcli /cx[/ex]/sx download mode=F offline [delay=*<value>*]

These commands support the drive firmware download using `Mode E` and `Mode F`. The mode options specify the SCSI WRITE BUFFER mode.

The description follows:

- `Mode E` – Downloads the microcode and allows you to issue this command for multiple devices. You can only use this in an offline mode.
- `Mode F` – Activates the deferred microcode and allows you to issue this command to all devices in a safe manner. You can only use this in an offline mode. You cannot issue this command before issuing the `Mode E` command. The default delay time is `15` seconds. You can specify any delay time between `1` to `300` seconds.

> **NOTE**
>
> You can download as well as activate the drive firmware by executing the `activatenow` command in the same command line. You can also specify the delay time, but the delay time specified by you is applicable only for activation and not for downloading the drive firmware.

Input examples for `Mode E` :

```
storcli /c0/e0/s0 download src=file.rom mode=E offline
```

Download successful.

```
storcli /c0/e0/sall download src=file.rom mode=E offline
```

Downloaded sequentially on the drives.

Input examples for `Mode F` :

```
storcli /c0/e0/sall download mode=F offline
```

Activation of the microcode successful

```
storcli /c0/e0/sall download mode=F offline delay=15
```

Activation completed with a 15-second delay.

Drive Firmware Update through Parallel HDD Microcode

MegaRAID provides an interface to update the drive firmware in both online and offline modes through host applications, such as StorCLI. Using the parallel HDD microcode update feature, firmware updates can be done simultaneously on multiple HDDs of the same family in an online mode. Also, the parallel HDD microcode update overcomes the VD tolerance level. You can use the parallel HDD microcode update feature to update up to eight devices at the same time. It is recommended to perform the parallel HDD microcode update in system maintenance mode.

The parallel HDD microcode update is not supported in the following scenarios:

- If physical drive firmware download is already in progress on any physical drive.
- If Pinned Cache is present on the controller.
- Online firmware upgrade is not supported if `FEATURE SET` value is enabled for `DEFAULT` and disabled for `LOW COST` .

Command Usage Examples
```
storcli /c0/ex/sall download src=drv_fw.lod [mode=5/7] [parallel] [force]
storcli /c1/e1/sall download src=drivefirmware.lod mode=5 parallel
```

Where:

- `c` – Controller number
- `x` – The index of either the controller or the enclosure

- `e` – Enclosure number
- `s` – Slot number
- `sall` – All drives
- `parallel` – Indicates firmware update is done in parallel mode
- `force` – Indicates whether you want to force this operation

storcli /c0/e1/sall download status

This command provides the current firmware download status on the specified drive list.

Locate Drives Commands

The StorCLI utility supports the following commands to locate a drive and activate the physical disk activity LED:

```
storcli /cx[/ex]/sx start locate
storcli /cx[/ex]/sx stop locate
```

The detailed description for each command follows.

storcli /cx[/ex]/sx start locate

This command locates a drive and activates the drive's LED.

Input example:

```
storcli /c0/e25/s4 start locate
```

storcli /cx[/ex]/sx stop locate

This command stops a locate operation and deactivates the drive's LED.

Input example:

```
storcli /c0/e25/s4 stop locate
```

Prepare to Remove Drives Commands

The StorCLI utility supports the following commands to prepare the physical drive for removal:

```
storcli /cx/ex/sx spindown
storcli /cx/ex/sx spinup
```

The detailed description for each command follows.

storcli /cx/ex/sx spindown

This command spins down an unconfigured drive and prepares it for removal. The drive state is unaffiliated, and it is marked offline.

Input example:

```
storcli /cx/e25/s4 spindown
```

storcli /cx/ex/sx spinup

This command spins up a spun-down drive and the drive state is unconfigured good.

Input example:

```
storcli /c0/e25/s4 spinup
```

> **NOTE**
>
> The spinup command works on a physical drive only if the user had previously issued a spindown command on the same physical drive.

Drive Security Command

The StorCLI utility supports the following drive security commands:

```
storcli /cx[/ex]/sx show securitykey keyid
```

storcli /c*x*[/e*x*]/s*x* show securitykey keyid

This command shows the security key for secured physical drives.

Input example:

```
storcli /c0/[e25]/s4 show SecurityKey keyid
```

storcli /c*x*/[e*x*]/s*x* set security = on

This command sets the security on the FDE-capable JBOD drive.

Input example:

```
storcli /c0/[e25]/s4 set security = on
```

Drive Secure Erase Commands

The StorCLI Utility supports the following drive erase commands:

```
storcli /cx[/ex]/sx secureerase [force]
storcli /cx[/ex]/sx show erase
storcli /cx[/ex]/sx start erase [simple|normal|crypto|thorough] [patternA=<value1>] [patternB=<value2>]
storcli /cx[/ex]/sx stop erase
```

The detailed description for each command follows.

storcli /c*x*[/e*x*]/s*x* secureerase [force]

This command erases the drive's security configuration and securely erases data on a drive. You can use the `force` option as a confirmation to erase the data on the drive and the security information.

Input example:

```
storcli /c0/e25/s1 secureerase
```

> **NOTE**
> This command deletes data on the drive and the security configuration, and this data is no longer accessible. This command is used for SED drives only.

storcli /c*x*[/e*x*]/s*x* show erase

This command provides the status of erase operation on non-SED drives.

Input example:

```
storcli /c0/e25/s1 show erase
```

storcli /c*x*[/e*x*]/s*x* start erase [simple|normal|thorough|crypto|standard] [patternA=<*val1*>] [patternB=<*val2*>]

> **NOTE**
> The erase option is supported only on UG drives and is not supported on JBOD drives.

This command securely erases non-SED drives. The drive is written with erase patterns to make sure that the data is securely erased. You can use the following options with the `start erase` command.

Table 12: Drive Erase Command Options

Options	Value Range	Description
`erase`	`simple` : Single pass, single pattern write `normal` : Three pass, three pattern write `thorough` : Nine pass, repeats the normal write three times `crypto` : Performs cryptographic erase for SSD drives	Secures erase type.
`patternA`	8-bit value	Erases pattern A to overwrite the data.
`patternB`	8-bit value	Erases pattern B to overwrite the data.

Input example:

```
storcli /c0/e25/s1 start erase thorough patternA=10010011 patternB=11110000
```

Drive Sanitize Command

The StorCLI Utility supports the following drive sanitize commands:

```
storcli /cx[/ex]/sx start sanitize [cryptoerase| overwrite| blockErase ] [ause]
 storcli /cx[/ex]/sx show sanitize
```

storcli /cx[/ex]/sx start sanitize [cryptoerase| overwrite| blockErase] [ause]

This command lets you erase the data that resides on a physical drive. You can use the following sanitize-type arguments with the start sanitize command.

`cryptoerase` – This argument corrupts the encryption keys that may have been present on the drive.

`overwrite` – This argument overwrites all zeros to the data that may be present on the existing drives.

`blockerase` – This argument allows the drive to clear or erase the existing data drive.

`ause` – If, for some reason, the sanitize operation fails, the system tries to bring the drive out of the failure mode irrespective of whether the Allow Unrestricted Sanitize Exit (AUSE) argument is specified or not. However, if this argument is specified, and if the system succeeds in bringing the drive out of the failure mode, the drive is then returned as an Unconfigured Good drive. If you do not specify the `ause` argument, and if the sanitize operation fails, the system places the drive in an Unconfigured Bad state.

Example

```
storcli /c0/e0/s4 start sanitize overwrite
```

storcli /cx/[ex]/sx show sanitize

This command displays the progress of the sanitize operation in percentage.

xample:

```
storcli /c0/e0/s4 show sanitize
```

Rebuild Drives Commands

The following commands rebuild drives in the StorCLI utility:

```
storcli /cx[/ex]/sx pause rebuild
```

```
storcli /cx[/ex]/sx resume rebuild
storcli /cx[/ex]/sx show rebuild
storcli /cx[/ex]/sx start rebuild
storcli /cx[/ex]/sx stop rebuild
```

> **NOTE**
> If enclosures are used to connect physical drives to the controller, specify the enclosure ID in the command.

The detailed description for each command follows.

storcli /cx[/ex]/sx pause rebuild

This command pauses an ongoing rebuild process. You can run this command only for a drive that is currently rebuilding.

Input example:

```
storcli /c0/s4 pause rebuild
```

storcli /cx[/ex]/sx resume rebuild

This command resumes a paused rebuild process. You can run this command only when a paused rebuild process for the drive exists.

Input example:

```
storcli /c0/s4 resume rebuild
```

storcli /cx[/ex]/sx show rebuild

This command shows the progress of the rebuild process in percentage.

The estimated time (in minutes) left to complete the operation is also shown.

Input example:

```
storcli /c0/s4 show rebuild
```

storcli /cx[/ex]/sx start rebuild

This command starts a rebuild operation for a drive.

Input example:

```
storcli /c0/s4 start rebuild
```

storcli /cx[/ex]/sx stop rebuild

This command stops a rebuild operation. You can run this command only for a drive that is currently rebuilt.

Input example:

```
storcli /c0/s4/e4 stop rebuild
```

Drive Copyback Commands

The StorCLI Utility supports the following commands for drive copyback:

```
storcli /cx[/ex]/sx pause copyback
storcli /cx[/ex]/sx resume copyback
```

```
storcli /cx show copyback
storcli /cx[/ex]/sx start copyback target=eid:sid
storcli /cx[/ex]/sx stop copyback
storcli /cx set copyback=on type=ctrl
storcli /cx set copyback=on type=smarthdd
storcli /cx set copyback=on type=ssd
storcli /cx set copyback=on type=all
```

The detailed description for each command follows.

> **NOTE**
> In the copyback commands, `cx[/ex]/sx` indicates the source drive and `eid:sid` indicates the target drive.

> **NOTE**
> When a copyback operation is enabled, the alarm continues to beep even after a rebuild is complete; the alarm stops beeping only when the copyback operation is completed.

storcli /cx[/ex]/sx pause copyback

This command pauses a copyback operation. You can run this command only when a copyback operation is running.

Input example:

```
storcli /c0/e25/s4 pause copyback
```

storcli /cx[/ex]/sx resume copyback

This command resumes a paused copyback operation. You can run this command only when a paused copyback process exists for the drive.

Input example:

```
storcli /c0/e25/s4 resume copyback
```

storcli /cx show copyback

This command shows the progress of the copyback operation in percentage.

The estimated time (in minutes) left to complete the operation is also shown.

Input example:

```
storcli /c0/e25/s4 show copyback
```

storcli /cx[/ex]/sx start copyback target=eid:sid

This command starts a copyback operation for a drive.

Input example:

```
storcli /c0/e25/s4 start copyback target=25:8
```

storcli /cx[/ex]/sx stop copyback

This command stops a copyback operation. You can run this command only on drives that have the copyback operation running.

Input example:

```
storcli /c0/e25/s4 stop copyback
```

> **NOTE**
> A stopped rebuild process cannot be resumed.

storcli /c*x* set copyback=on type=ctrl

This command sets a control copyback operation.

Input example:

```
storcli /c0/e25/s4 set copyback type=ctrl
```

storcli /c*x* set copyback=on type=smarthdd

This command sets a smart HDD copyback operation.

Input example:

```
storcli /c0/e25/s4 show copyback type=smarthdd
```

storcli /c*x* set copyback=on type=smartssd

This command sets a smart SSD copyback operation.

Input example:

```
storcli /c0/e25/s4 show copyback type=smartssd
```

storcli /c*x* set copyback=on type=all

This command sets a copyback operation.

Input example:

```
storcli /c0/e25/s4 show copyback type=all
```

Hot Spare Drive Commands

The following commands create and delete hot spare drives:

```
storcli /cx[/ex]/sx add hotsparedrive
{dgs=<n|0,1,2...>}[enclaffinity][nonrevertible]
storcli /cx/[ex]/sx delete hotsparedrive
```

> **NOTE**
> If enclosures are connected to the physical drives of the controller, specify the enclosure ID in the command.

The detailed description for each command follows.

storcli /c*x*[/e*x*]/s*x* add hotsparedrive [{dgs=<*n*|0,1,2...>}] [enclaffinity][nonrevertible]

This command creates a hot spare drive. You can use the following options to create a hot spare drive.

Table 13: Add Hot Spare Drive Input Options

Option	Value Range	Description
dgs	Valid drive group number	Specifies the drive group to which the hot spare drive is dedicated.
enclaffinity	Valid enclosure number	Specifies the enclosure with which the hot spare is associated. If this option is specified, affinity is set; if it is not specified, there is no affinity. Affinity cannot be removed after it is set for a hot spare drive.
nonrevertible	—	Sets the drive as a nonrevertible hot spare.

Input example:

```
storcli /c0/e25/s4,5 add hotsparedrive
```

This command sets the drives `/c0/e25/s4,5` as global hot spare.

Input example:

```
storcli /c0/e25/s6,8 add hotsparedrive dgs=0,1
```

This command sets `/c0/e25/s6,8` as dedicated hot spare for disk groups 0,1.

storcli /cx/[ex]/sx delete hotsparedrive

This command deletes a hot spare drive.

Input example:

```
storcli /c0/e25/s4,5 delete hotsparedrive
```

Drive Performance Monitoring Commands

The StorCLI utility supports the following commands for drive performance monitoring:

```
storcli /cx show pdfailevents [lastoneday] [lastseqnum=<val>] [file=filename]
storcli /cx set pdfaileventoptions detectiontype=val correctiveaction=val errorrthreshold=val
```

The detailed description for each command follows.

storcli / cx show pdfailevents

This command shows all of the drive predictive failure events.

Input example:

```
storcli /c0 show pdfailevents
```

storcli / cx show pdfailevents lastoneday

This command shows all of the drive predictive failure events that occurred in the last 24 hours.

Input example:

```
storcli /c0 show pdfailevents lastoneday
```

storcli / c*x* show pdfailevents lastseqnum=x*x*]

This command shows all of the drive predictive failure events generated from the specified sequence number.

Input example:

```
storcli /c0 show pdfailevents lastseqnum=10
```

storcli / c*x* set pdfaileventoptions detectiontype=*val* correctiveaction=*val* errorrthreshold=*val*

This command provides the current settings of the `pdfaileventoptions` set on the controller and the various options to change these settings.

Input example 1:

```
storcli /c0 set pdfaileventoptions detectiontype=x
```

Where:

- `0` – Detection disabled
- `1` – Detection enabled, high latency for reads is OK
- `2` – Detection enabled, aggressive (high latency for reads is not OK)
- `3` – Detection enabled, use NVDATA specified value, see `recoveryTimeLimit` and `writeRetryCount`

This command sets the detection type for the drive. The valid range is 0 to 3.

> **NOTE**
>
> For the changes to take effect, a reboot is required.

Input example 2:

```
storcli /c0 set pdfaileventoptions correctiveaction=x
```

Where:

- `0` – Only log events
- `1` – Log events, take corrective action based on SMARTer.

This command sets the corrective actions to be taken when the media error is detected. The valid value is 0 or 1.

Input example 3:

```
storcli /c0 set pdfaileventoptions errorrthreshold=x
```

Where:

- `0` = `1` – One error every 8 hours (least tolerant)
- `1` = `8` – One error every 1 hour
- `2` = `32` – One error every 15 minutes
- `3` = `90` – One error every 5 minutes (most tolerant of drive with degraded media)

This command sets the error threshold for the controller. The valid range is 0 to 3.

storcli / c*x* show dpm

This command shows the drive performance monitoring status on the controller.

Input example:

```
storcli /c0 show dpm
```

storcli / c*x* start dpm

This command starts the performance monitoring on the controller.

Input example:

```
storcli /c0 start dpm
```

storcli / c*x* stop dpm

This command stops the performance monitoring on the controller.

Input example:

```
storcli /c0 stop dpm
```

storcli /c*x* delete dpmstat type = HIST | LCT | RA | EXT | All

This command deletes the drive performance monitoring statistics of the controller.

Where:

- `HIST` – Historgram of response time.
- `LCT` – Long Time commands.
- `RA` – Running average drive statistics.
- `EXT` – Extended DPM information.

storcli /c*x*[/e*x*]/s*x* show dpmstat type = HIST | LCT | RA | EXT [logfile[=filename]]

This command shows the drive performance monitoring statistics of the mentioned drive.

Where:

- `HIST` – Historgram of response time.
- `LCT` – Long Time commands.
- `RA` – Running average drive statistics.
- `EXT` – Extended DPM information.
- `logfile` – If the file name is not specified, it logs to `starsas.log` or to a user-specified file.

Virtual Drive Commands

The StorCLI utility supports the following virtual drive commands. The following table describes frequently used virtual drive commands.

Table 14: Virtual Drives Commands Quick Reference Table

Commands	Value Range	Description
`add`	See Table 15, Add RAID Configuration Input Options and Table 16, Add RAID Configuration Input Options.	Creates virtual drives.
`delete`	`cc` or `cachecade` : Deletes CacheCade virtual drives. `force` : Deletes the virtual drive where operating system is present.	Deletes a virtual drive.
`set`	See Table 15, Add RAID Configuration Input Options, Table 16, Add RAID Configuration Input Options, and Change Virtual Properties Commands.	Sets virtual drive properties.

Commands	Value Range	Description
show	all : Shows all properties of the virtual drive. cc : Shows properties of CacheCade virtual drives. See Virtual Drive Show Commands.	Shows virtual drive information.

Add Virtual Drives Commands

The StorCLI Utility supports the following commands to add virtual drives:

```
storcli /cx add vd raid[0|1|5|6|00|10|50|60][Size=<VD1_Sz>,<VD2_Sz>,..|remaining] [name=<VDNAME1>,..]
 drives=e:s|e:s-x,y,e:s-x,y,z [PDperArray=x][SED] [pdcache=on|off|default][pi] [DimmerSwitch(ds)=default|
automatic(auto)| none|maximum(max)|MaximumWithoutCaching(maxnocache)]
[wt|wb|awb] [nora|ra] [direct|cached][cachevd] [Strip=<8|16|32|64|128|256|1024>] [AfterVd=X][EmulationType=0|
1|2] [Spares = [e:]s|[e:]s-x|[e:]s-x,y] [force][ExclusiveAccess][Cbsize=0|1|2 Cbmode=0|1|2|3|4|7]
```

> **NOTE**
>
> The supported strip size can vary from a minimum of 64 KB to 1 MB for MegaRAID controllers and only 64 KB
> for Integrated MegaRAID controllers.

```
storcli /cx add vd each raid0 [name=<VDNAME1>,..] [drives=e:s|e:s-x|e:s-x,y] [SED] [pdcache=on|off|default]
[pi] [DimmerSwitch(ds)=default|automatic(auto)| none|maximum(max)|MaximumWithoutCaching(maxnocache)] [wt|wb|
awb] [nora|ra] [direct|cached][EmulationType=0|1|2] [Strip=<8|16|32|64|128|256|1024>][ExclusiveAccess]
```

> **NOTE**
>
> The supported strip size can vary from a minimum of 64 KB to 1 MB for MegaRAID controllers and only 64 KB
> for Integrated MegaRAID controllers.

```
storcli /cx add VD cachecade|cc raid[0,1] drives = [e:]s|[e:]s-x|[e:]s-x,y [WT|WB] [assignvds = 0,1,2]
```

This command creates a RAID configuration. You can use the options in the following table to create the RAID volume.

> **NOTE**
> * indicates default values in the following commands.

The detailed description for each command follows.

**storcli /cx add vd raid[0|1|5|6|00|10|50|60][Size=<VD1_Sz>,<VD2_Sz>,..|*remaining] [name=<VDNAME1>,..]
drives=e:s|e:s-x|e:s-x,y,e:s-x,y,z [PDperArray=x][SED] [pdcache=on|off|*default][pi] [DimmerSwitch(ds)=default|
automatic(auto)| *none|maximum(max)|MaximumWithoutCaching(maxnocache)][cachevd][ExclusiveAccess|
SharedAccess*]** [wt|*wb |awb] [nora|*ra] [*direct|cached] [EmulationType=0][Strip=<8|16|32|64|128|256|1024>]
[AfterVd=X] [Spares = [e:]s|[e:]s-x|[e:]s-x,y] [force]**

> **NOTE**
>
> The supported strip size can vary from a minimum of 64 KB to 1 MB for MegaRAID controllers and only 64 KB
> for Integrated MegaRAID controllers.

Table 15: Add RAID Configuration Input Options

Option	Value Range	Description
raid	[0|1|5|6|00|10|50|60].	Sets the RAID type of the configuration.
size	Maximum size based on the physical drives and RAID level.	Sets the size of each virtual drive. The default value is for the capacity of all referenced disks.

Option	Value Range	Description
name	15 characters in length.	Specifies the drive name for each virtual drive.
drives	Valid enclosure number and valid slot numbers for the enclosure.	In $e:s\|e:s-x\|e:s-x,y$: • e specifies the enclosure ID. • s represents the slot in the enclosure. • $e:s-x$ is the range convention used to represent slots s to x in the enclosure e (250 characters maximum). Make sure that the same block size (in a physical drive) is used in each [$e:s$] pair. As an example, if you use 4096 bytes in the $e0:s0$ pair, use 4096 bytes in the $e1:s1$ pair too. Mixing of block sizes between the [$e:s$] pairs is not supported.
pdperarray	1–16.	Specifies the number of physical drives per array. The $default$ value is automatically chosen.
sed	—	Creates security-enabled drives.
pdcache	on|off|default.	Enables or disables PD cache.
pi	—	Enables protection information.
dimmerswitch	default : Logical device uses controller default power-saving policy. automatic (auto): Logical device power savings are managed by firmware. none : No power-saving policy. maximum (max): Logical device uses maximum power savings. MaximumWithoutCaching (maxnocache): Logical device does not cache write to maximize power savings.	Specifies the power-saving policy. Sets to default automatically.
direct|cached	cached : Cached I/O. direct : Direct I/O.	Sets the logical drive cache policy. Direct I/O is the default.
EmulationType	0 : Default emulation, which means if there are any 512e drives in the configured ID, then the physical bytes per sector are shown as 512e (4k). If there are no 512e drives, the physical bytes per sector will be 512n. 1 : Disable, which means even though there are 512e drives in the configured ID, the physical bytes per sector will be shown 512n. 2 : Force, which means even though there are no 512e drives in the configured ID, the physical bytes per sector will be shown as 512e (4k).	
wt|wb|awb	• wt : Write through. • wb : Write back. • awb : Always Write Back.	Enables write through. Write back is the default.
nora|ra	• ra : Read ahead. • nora : No read ahead.	Disables read ahead. Enabled is the default.
cachevd	—	Enables SSD caching on the created virtual drive.

Option	Value Range	Description
strip	8,16,32,64,128,256,512,1024. **Note:** The supported strip size can vary from a minimum of 64 KB to 1 MB for MegaRAID controllers and only 64 KB for Integrated MegaRAID controllers.	Sets the strip size for the RAID configuration.
aftervd	Valid virtual drive number.	Creates the VD in the adjacent free slot next to the specified VD.
spares	Number of spare physical drives present.	Specifies the physical drives that are to be assigned to a disk group for spares.
force	Forces the addition of a secured physical drive to an existing drive group without security.	Forces a security-capable physical drive to be added to a drive group without security.

Input example:

```
storcli /c0 add vd raid10 size=2gb,3gb,4gb names=tmp1,tmp2,tmp3 drives=252:2-3,5,7 pdperarray=2
```

storcli /cx add vd cc|cachecade raid[0,1,10] drives=[e:]s|[e:]s-x|[e:]s-x,y [[wt|*wb|awb]] [assignvds=0,1,2]

This command creates CacheCade virtual drives and associates existing virtual drives to CacheCade virtual drives. You can use the following options to create the CacheCade virtual drive.

Table 16: Add RAID Configuration Input Options

Option	Value Range	Description
cachecade	—	Creates a CacheCade virtual drive.
raid	0,1,10.	Sets the RAID type of the CacheCade virtual drive.
drives	Valid enclosure number and valid slot number	See the drives row in the previous table for format.
wt|*wb|awb	wt : Enables write through. wb : Enables write back. awb Enables always write back.	Enables or disables write cache.
assignvds	Valid virtual drive number (0 to 63).	Specifies the list of virtual drives associated with the new CacheCade virtual drives.

Input example:

```
storcli /c0 add vd raid10 size=2gb,3gb,4gb names=tmp1,tmp2,tmp3 drives=252:2-3, 7
```

Delete Virtual Drives Commands

The StorCLI utility supports the following virtual drive delete commands:

```
storcli /cx/vx|vall del
storcli /cx/vx|vall del cachecade
storcli /cx/vx|vall del force
```

```
storcli /cx/vx del [cachecade] [discardcache] [force]
```

>   **NOTE**
>
>   If the virtual drive has user data, you must use the `force` option to delete the virtual drive.
>
>   A virtual drive with a valid master boot record (MBR) and a partition table is considered to contain user data.

If you delete a virtual drive with a valid MBR without erasing the data and then create a new virtual drive using the same set of physical drives and the same RAID level as the deleted virtual drive, the old unerased MBR still exists at block 0 of the new virtual drive, which makes it a virtual drive with valid user data. Therefore, you must provide the `force` option to delete this newly created virtual drive.

The detailed description for each command follows.

storcli /cx/vx|vall del

This command deletes a particular virtual drive, or when the `vall` option is used, all the virtual drives on the controller are deleted.

Input example:

```
storcli /c0/v2 del
```

>   **ATTENTION**
>
>   This command deletes virtual drives. Data located on these drives will no longer be accessible.

storcli /cx/vx|vall del cachecade

This command deletes a specific CacheCade virtual drive on a controller or all of the CacheCade configuration for a controller.

Input example:

```
storcli /c0/vall del cachecade
```

>   **ATTENTION**
>
>   This command deletes virtual drives. Data located on these drives will no longer be accessible.

storcli /cx/vx|vall del force

This command deletes a virtual drive only after the cache flush is completed. With the `force` option, the command deletes a virtual drive without waiting for the cache flush to complete.

Input example:

```
storcli /c0/v2 del force
```

>   **ATTENTION**
>
>   This command deletes the virtual drive where the operating system is present. Data located on these drives and the operating system of the drive will no longer be accessible.

storcli /cx/vx del [cachecade] [discardcache] [force]

This command with the `discardCache` option deletes the virtual drive without flushing the cached data.

Input example:

```
storcli /c0/v2 delete discardcache
```

Virtual Drive Show Commands

The StorCLI utility supports the following virtual drive show commands:

```
storcli /cx/vx show
storcli /cx/vx show all [logfile[=filename]]
```

The detailed description for each command follows.

storcli /cx/vx show

This command shows the summary of the virtual drive information.

Input example:

```
storcli /c0/v0 show
```

storcli /cx/vx show all [logfile[=filename]]

The `show all` command shows all of the virtual drive information, which includes the virtual drive information, physical drives used for the virtual drives, and virtual drive properties.

If you use the `logfile` option in the command syntax, the logs are written to the specified file. If you do not specify a file name, then the logs are written to the `storsas.log` file. If you do not use the `logfile` option in the command syntax, the entire log output is printed to the console.

Input example:

```
storcli /c0/v0 show all logfile = log.txt
```

Preserved Cache Commands

If a virtual drive becomes offline or is deleted because of missing physical disks, the controller preserves the dirty cache from the virtual disk. The StorCLI utility supports the following commands for preserved cache:

```
storcli /cx/vx delete preservedCache [force]
storcli /cx show preservedCache
```

The detailed description for each command follows.

storcli /cx/vx delete preservedcache

This command deletes the preserved cache for a particular virtual drive on the controller in missing state. Use the `force` option to delete the preserved cache along with the virtual drive when the virtual drive is in an offline state.

Input example:

```
storcli /c0/v1 delete preservedcache
```

storcli /cx show preservedCache

This command shows the virtual drive that has preserved cache and whether the virtual drive is offline or missing.

Input example:

```
storcli /c0 show preservedCache
```

Change Virtual Properties Commands

The StorCLI utility supports the following commands to change virtual drive properties:

```
storcli /cx/vx set accesspolicy=<rw|ro|blocked|rmvblkd>
storcli /cx/vx set iopolicy=<cached|direct>
storcli /cx/vx set name=<namestring>
storcli /cx/vx set pdcache=<on|off|default>
storcli /cx/vx set rdcache=<ra|nora>
storcli /cx/vx|vall set ssdcaching=<on|off>
storcli /cx/vx|vall set HostAccess=ExclusiveAccess|SharedAccess
storcli /cx/vx set wrcache=<wt|wb|awb>
storcli /cx/vx set emulationType=0|1|2
storcli /cx/vx set ds=Default|Auto|None|Max|MaxNoCache
storcli /cx/vx set autobgi=On|Off
storcli /cx/vx set pi=Off
storcli /cx/vx set bootdrive=<On|Off>
storcli /cx/vx set hidden=On|Off
storcli /cx/vx set cbsize=0|1|2 cbmode=0|1|2|3|4|7
```

The detailed description for each command follows.

storcli /cx/vx set accesspolicy=<rw|ro|blocked|rmvblkd>

This command sets the access policy on a virtual drive to read/write, read only, blocked, or rmvblkd (remove or blocked).

Input example:

```
storcli /c0/v0 set accesspolicy=rw
```

storcli /cx/vx set iopolicy=<cached|direct>

This command sets the I/O policy on a virtual drive to cached I/O or direct I/O.

Input example:

```
storcli /c0/v0 set iopolicy=cached
```

storcli /cx/vx set name=<namestring>

This command names a virtual drive. The name is restricted to 15 characters.

Input example:

```
storcli /c0/v0 set name=testdrive123
```

storcli /cx/vx set pdcache=<on|off|default>

This command sets the current disk cache policy on a virtual drive to on, off, or default setting.

Input example:

```
storcli /c0/v0 set pdcache=on
```

storcli /c*x*/v*x* set rdcache=<ra|nora>

This command sets the read cache policy on a virtual drive to read ahead or no read ahead.

Input example:

```
storcli /c0/v0 set rdcache=nora
```

storcli /c*x*/v*x*|vall set ssdcaching=<on|off>

This command assigns CacheCade virtual drives. If `ssdcaching=off` , the CacheCade virtual drive is removed.

Input example:

```
storcli /c0/v0 set ssdcaching=on
```

storcli /c*x*/v*x*|vall set HostAccess=ExclusiveAccess|SharedAccess

This command sets the host access policy for the virtual drive. When the host access policy is exclusive access, a server has exclusive access to the virtual drive. The virtual drive cannot be shared between servers. If the host policy is shared access, the virtual drive can be shared between servers.

Input example:

```
storcli /c0/v0 set HostAccess=ExclusiveAccess
```

storcli /c*x*/v*x* set wrcache=<wt|wb|awb>

This command sets the write cache policy on a virtual drive to write back, write through, or always write back.

Input example:

```
storcli /c0/v0 set wrcache=wt
```

storcli /c*x*/v*x* set hidden=on|off

This command hides or unhides a virtual drive. If `hidden=on` , the virtual drive is not exported to the OS. The OS will not be able to read or write to that virtual drive until the virtual drive is unhidden.

Input example:

```
storcli /c0/v0 set hidden=on
```

storcli /c*x*/v*x* set cbsize=0|1|2 cbmode=0|1|2|3|4|7

This command sets the Cache bypass size and the Cache bypass mode on a virtual drive.

The values for the `cbsize` options follow:

- `0` – 64k cache bypass.
- `1` – 128k cache bypass.
- `2` – 256k cache bypass.

The values for the `cbmode` options follow:

- `0` – Enable the intelligent mode cache bypass.
- `1` – Enable the standard mode cache bypass.
- `2` – Enable the custom mode cache bypass 1.
- `3` – Enable the custom mode cache bypass 2.
- `4` – Enable the custom mode cache bypass 3.
- `7` – Disable Cache bypass.

> **NOTE**
>
> When `cbmode` is set to 7, the user supplied `cbsize` value is ignored.

Input example:

```
storcli /c0/v0 set cbsize=1 cbmode=2
```

Virtual Drive Initialization Commands

The StorCLI utility supports the following commands to initialize virtual drives:

```
storcli /cx/vx show init
storcli /cx/vx start init [full][Force]
storcli /cx/vx stop init
```

> **NOTE**
>
> If the virtual drive has user data, you must use the `force` option to initialize the virtual drive.
>
> A virtual drive with a valid MBR and partition table is considered to contain user data.

The detailed description for each command follows.

storcli /c*x*/v*x* show init

This command shows the initialization progress of a virtual drive in percentage.

The estimated time (in minutes) left to complete the operation is also shown.

Input example:

```
storcli /c0/v2 show init
```

storcli /c*x*/v*x* start init [full]

This command starts the initialization of a virtual drive. The default initialization type is fast initialization. If the `full` option is specified, full initialization of the virtual drive starts.

Input example:

```
storcli /cx/vx start init full
```

storcli /c*x*/v*x* stop init

This command stops the initialization of a virtual drive. A stopped initialization cannot be resumed.

Input example:

```
storcli /c0/v0 stop init
```

Virtual Drive Erase Commands

The StorCLI Utility supports the following commands to erase virtual drives:

```
storcli /cx/vx start erase
storcli /cx/vx show erase
```

The detailed description for each command follows.

storcli /cx/vx start erase

This command starts the data erase operation on the specified virtual drive.

Input example:

```
storcli /c0/v0 start erase
```

storcli /cx/vx show erase

This command shows the status of the erase operation on the virtual drive.

Input example:

```
storcli /c0/v0 show erase
```

Virtual Drive Migration Commands

> **NOTE**
> The virtual drive migration commands are not supported in Embedded MegaRAID.

The StorCLI Utility supports the following commands for virtual drive migration (reconstruction):

```
storcli /cx/vx show migrate
storcli /cx/vx start migrate <type=raidx> [option=<add|remove> drives=[e:]s|[e:]s-x|[e:]s-x,y] [Force]
```

> **NOTE**
> Usable Capacity Reduction is not possible without a RAID level migration.

The detailed description for each command follows.

storcli /cx/vx show migrate

This command shows the progress of the virtual drive migrate operation in percentage.

The estimated time (in minutes) left to complete the operation is also shown.

Input example:

```
storcli /c0/v0 show migrate
```

storcli /cx/vx start migrate <type=raidlevel> [option=<add | remove> drives=<e1:s1,e2:s2 ...>]

This command starts the reconstruction on a virtual drive to the specified RAID level by adding or removing drives from the existing virtual drive. You can use the following options with the `start migrate` command.

Table 17: Virtual Drive Migration Command Options

Options	Value Range	Description
`type = RAID level`	`RAID [0｜1｜5｜6]`	The RAID level to which the virtual drive must be migrated.
`[option=<add ｜ remove> drives=<e1:s1,e2:s2, … >]`	`add` : Adds drives to the virtual drive and starts reconstruction. `remove` : Removes drives from the virtual drive and starts reconstruction. `drives` : The enclosure number and the slot number of the drives to be added to the virtual drive. Make sure that the same block size (in a physical drive) is used in each [*e:s*] pair. As an example, if you use 4096 bytes in the *e0:s0* pair, use 4096 bytes in the *e1:s1* pair too. Mixing of block sizes between the [*e:s*] pairs is not supported.	Adds or removes drives from the virtual drive.

Virtual drive migration can be done between the following RAID levels.

Table 18: Virtual Drive Migration Table

Initial RAID level	Migrated RAID level
RAID 0	RAID 1
RAID 0	RAID 5
RAID 0	RAID 6
RAID 1	RAID 0
RAID 1	RAID 5
RAID 1	RAID 6
RAID 5	RAID 0
RAID 5	RAID 6
RAID 6	RAID 0
RAID 6	RAID 5

Input example:

In the following example, 252 is the enclosure number and 0 , 1 , and 2 are the slot numbers.

```
storcli/c0/v0 start migrate type=raid0 option=add drives=252:0,252:1,252:2
```

Virtual Drive Consistency Check Commands

The StorCLI utility supports the following commands for virtual drive consistency checks:

```
storcli /cx/vx pause cc
storcli /cx/vx resume cc
storcli /cx/vx show cc
storcli /cx/vx start cc [force]
```

```
storcli /cx/vx stop cc
```

> **NOTE**
> If enclosures are used to connect the physical drives to the controller, specify the IDs in the command.

The detailed description for each command follows.

storcli /c*x*/v*x* pause cc

This command pauses an ongoing consistency check process. You can resume the consistency check at a later time. You can run this command only on a virtual drive that has a consistency check operation running.

Input example:

```
storcli /c0/v4 pause cc
```

storcli /c*x*/v*x* resume cc

This command resumes a suspended consistency check operation. You can run this command on a virtual drive that has a paused consistency check operation.

Input example:

```
storcli /c0/v4 resume cc
```

storcli /c*x*/v*x* show cc

This command shows the progress of the consistency check operation in percentage.

The estimated time (in minutes) left to complete the operation is also shown.

Input example:

```
storcli /c0/v5 show cc
```

storcli /c*x*/v*x* start cc force

This command starts a consistency check operation for a virtual drive. Typically, a consistency check operation is run on an initialized virtual drive. Use the `force` option to run a consistency check on an uninitialized drive.

Input example:

```
storcli /c0/v4 start cc
```

storcli /c*x*/v*x* stop cc

This command stops a consistency check operation. You can run this command only for a virtual drive that has a consistency check operation running.

Input example:

```
storcli /c0/v4 stop cc
```

> **NOTE**
> You cannot resume a stopped consistency check process.

Background Initialization Commands

The StorCLI utility supports the following commands for background initialization:

```
storcli /cx/vx resume bgi
storcli /cx/vx set autobgi=<on|off>
storcli /cx/vx show autobgi
storcli /cx/vx show bgi
storcli /cx/vx stop bgi
storcli /cx/vx suspend bgi
```

The detailed description for each command follows.

storcli /cx/vx resume bgi

This command resumes a suspended background initialization operation.

Input example:

```
storcli /c0/v0 resume bgi
```

storcli /cx/vx set autobgi=<on|off>

This command sets the auto background initialization setting for a virtual drive to on or off.

Input example:

```
storcli /c0/v0 set autobgi=on
```

storcli /cx/vx show autobgi

This command shows the background initialization setting for a virtual drive.

The estimated time (in minutes) left to complete the operation is also shown.

Input example:

```
storcli /c0/v0 show autobgi
```

storcli /cx/vx show bgi

This command shows the background initialization progress on the specified virtual drive in percentage.

The estimated time (in minutes) left to complete the operation is also shown.

Input example:

```
storcli /c0/v0 show bgi
```

storcli /cx/vx stop bgi

This command stops a background initialization operation. You can run this command only for a virtual drive that is currently initialized.

Input example:

```
storcli /c0/v4 stop bgi
```

storcli /cx/vx pause bgi

This command suspends a background initialization operation. You can run this command only for a virtual drive that is currently initialized.

Input example:

```
storcli /c0/v4 pause bgi
```

Virtual Drive Expansion Commands

The StorCLI Utility supports the following commands for virtual drive expansion:

```
storcli /cx/vx expand size=<value> [expandarray]
storcli /cx/vx|vall show expansion
```

The detailed description for each command follows.

storcli /cx/vx expand size=<value> [expandarray]

This command expands the virtual drive within the existing array or if you replace the drives with drives larger than the size of the existing array. Although the value provided might be in MB, the value of the expanded size is displayed based on the nearest possible unit. Depending on the input (value) provided by you, storcli recognizes the size from the input provided by you and rounds up the size to the nearest percentage of free space remaining on the drive group; hence, the actual expanded size can differ from the size requested by you. If the expandarray option is specified, the existing array is expanded. If this option is not specified, the virtual drive is expanded.

storcli /cx/vx show expansion

This command shows the expansion information on the virtual drive with and without array expansion.

Input example:

```
storcli /c0/v0 show expansion
```

Display the Bad Block Table

The StorCLI Utility supports the following command to check for bad block entries of virtual drives on the selected controller:

```
storcli /cx/vx show bbmt
```

Input example:

```
storcli /c0/v0 show bbmt
```

Clear the LDBBM Table Entries

The StorCLI Utility supports the following command to clear the LDBBM table entries:

```
storcli /cx/vx delete bbmt
```

Input example:

```
storcli /c0/v0 delete bbmt
```

JBOD Commands

The StorCLI utility supports the switching behavior within the JBOD personality mode. StorCLI also supports configuration parameters for a personality and allows you to create and configure JBODs. You can create JBODs from all Unconfigured

Good drives or specific Unconfigured Good drives. You can also delete these JBODs. You can also choose JBOD as a boot device.

The Storage Command Line Interface utility supports the following JBOD commands:

```
storcli /cx/ex/sx set jbod

storcli /cx/eall/sall show jbod [all]

storcli /cx/ex/sx start init

storcli /cx/ex/sx stop init

storcli /cx/ex/sx start erase

storcli /cx/ex/sx stop erase

storcli /cx/ex/sx set bootdrive=<on|off>

storcli /cx/ex/sall delete jbod
```

For more information, see also *set personality behavior* under Show and Set Controller Properties Commands.

Create JBOD Manually

The StorCLI Utility has the option to convert all specified Unconfigured Good drives as JBODs.

> **NOTE**
> The drive token is optional. If you specify the drives, the JBODs are created on those specified drives; otherwise, StorCLI Utility creates JBODs on all available Unconfigured Good drives on the controller.

storcli /*cx*/*ex*/*sx* set jbod

This command allows you to add JBOD drive.

Input example:

```
storcli /c0/ex/sX set jbod
```

JBOD Properties

storcli /*cx*/eall/sall show jbod [all]

Input example:
```
storcli /c0/eall/sall show jbod
```

Table 19: Example Output of Available JBODs on the Controller

ID	EID:SLT	DID	State	Intf	Med	Size	SeSz	Model	Vendor	Port
0	10:01	2	Online	SAS	HDD	100GB	512B	ST91000640SS	SAMSUNG	0–3
1	10:03	5	Online	SAS	HDD	123GB	4K	ST91000640SS	SAMSUNG	0–3
2	10:04	6	Online	SAS	HDD	100GB	512B	ST91000640SS	SAMSUNG	0–3

JBOD Operations

JBODs can start and stop the INIT, and also erase operations on them. JBODs can also be set as a boot volume. Use the same drive commands for the respective operation to start and stop INIT on JBODs as follows.

storcli /c*x*/e*x*/s*x* start init

This command starts the initialization of a JBOD drive.

Input example:

```
storcli /c0/e10/s1 start init
```

storcli /c*x*/e*x*/s*x* show init

This command displays the initialization status.

Input example:

```
storcli /c0/e10/s0 show init
```

storcli /c*x*/e*x*/s*x* stop init

This command stops the initialization of a JBOD physical drive. A stopped initialization cannot be resumed.

Input example:

```
storcli /c0/e10/s0 stop init
```

storcli /c*x*/e*x*/s*x* start erase [simple | normal | thorough] [patternA=<*val*>] [patternB=<*val*>]

This command allows you to securely erase non-SED drives with the specified erase patterns.The drive is written with erase patterns to make sure that the data is securely erased. You can use the following options with the start erase command.

Table 20: Drive Erase Command Options

Options	Value Range	Description
erase	• simple : Single pass, single pattern write • normal : Three pass, three pattern write • thorough : Nine pass, repeats the normal write three times	Secure erase type.
patternA	8-bit value	Erase pattern A to overwrite the data.
patternB	8-bit value	Erase pattern B to overwrite the data.

Input example:

```
storcli /c0/e10/s0 start erase through patternA=10010011 patternB=11110000
```

storcli /c*x*/e*x*/s*x* show erase

This command displays the erase status.

Input example:

```
storcli /c0/e10/s0 show erase
```

storcli /c*x*/e*x*/s*xx* stop erase

This command stops the erase operation of a JBOD physical drive.

Input example:

```
storcli /c0/e10/s0 stop erase
```

storcli /cx/ex/sx set bootdrive= on|off

This command allows you to set the selected JBOD as boot volume.

Input example:

```
storcli /c0/e10/s0 set bootdrive= on|off
```

Delete JBODs or Volumes

To delete JBODs, use the drive's JBOD delete command.

storcli /cx/ex/sx delete jbod

This command deletes the specified JBOD.

Input example:

```
storcli /c0/e10/s0,1 delete jbod
```

storcli /cx/vall delete

This command deletes all the volumes on the controller.

Input example:

```
storcli /c0/vall delete
```

To delete all volumes on the controller, use the `vall delete` command.

Foreign Configuration Commands

The StorCLI utility supports the following commands to view, import, and delete foreign configurations:

```
storcli /cx/fall del|delete [securitykey=sssssssssss
storcli /cx/fall import [preview][securitykey=sssssssssss]
storcli /cx/fall show [all] [securitykey=sssssssssss]
```

> **NOTE**
> Provide the security key when importing a locked foreign configuration created in a different machine that is encrypted with a security key.

The detailed description for each command follows.

storcli /cx/fall|del| delete [securitykey=sssssssssss]

This command deletes the foreign configuration of a controller. Input the security key if the controller is secured.

Input example:

```
storcli /c0/fall delete
```

storcli /cx/fall import [preview] [securitykey=sssssssssss]

This command imports the foreign configurations of a controller. The `preview` option shows a summary of the foreign configuration before importing it.

Input example:

```
storcli /c0/fall import
```

storcli /cx/fall show [all][securitykey=sssssssssss]

This command shows the summary of the entire foreign configuration for a particular controller. The `all` option shows all the information of the entire foreign configuration.

> **NOTE**
> The EID:Slot column is populated for the foreign PDs that are locked.

Input example:

```
storcli /c0/fall show preview
storcli /c0/fall import preview
storcli /c0/fall show all
```

BIOS-Related Commands

The StorCLI Utility supports the following BIOS commands:

```
storcli /cx set bios [state=<on|off>] [Mode=<SOE|PE|IE|SME>] [abs=<on|off>]  [DeviceExposure=<value>]
```

The detailed description for the command follows.

storcli /cx set bios [state=<on|off>] [Mode=<SOE|PE|IE|SME>] [abs=<on|off>] [DeviceExposure=<value>]

This command enables or disables the MegaRAID controller's BIOS, sets the BIOS boot mode, and enables the BIOS to select the best logical drive as the boot drive. The mode options abbreviations follow:

- SOE: Stop on Errors.
- PE: Pause on Errors.
- IE: Ignore Errors.
- SME: Safe mode on Errors.

> **NOTE**
> The legacy BIOS can load a limited number of the PCI device's BIOS. Disable the MegaRAID BIOS to avoid issues during POST.

Input example:

```
storcli /c0 set bios state=on Mode=SOE abs=on deviceexposure=20
```

OPROM BIOS Commands

The StorCLI Utility supports the following OPROM BIOS commands:

```
storcli /cx/ex/sx set bootdrive=on|off
storcli /cx/vx set bootdrive=on|off
storcli /cx show bootdrive
```

The detailed description for each command follows.

storcli /cx/ex/sx set bootdrive=on|off

This command sets the specified physical drive as the boot drive. During the next reboot, the BIOS looks for a boot sector in the specified physical drive.

Input example:

```
storcli /c0/e32/s4 set bootdrive=on
```

storcli /*cx*/*vx* set bootdrive=on|off

This command sets the specified virtual drive as the boot drive. During the next reboot, the BIOS looks for a boot sector in the specified virtual drive.

Input example:

```
storcli /c0/v0 set bootdrive=on
```

storcli/c*x*/v*x* show bootdrive

This command shows the boot drive for the controller. The boot drive can be a physical drive or a virtual drive.

Input example:

```
storcli /c0/v0 show bootdrive
```

Drive Group Commands

This section describes the drive group commands.

Drive Group Show Commands

The StorCLI utility supports the following drive group commands:

```
storcli /cx/dall show
storcli /cx/dall show all
storcli /cx/dall show cachecade
storcli /cx/dx show
storcli /cx/dx show all
storcli /cx/dx set security=on
storcli /cx/dx split mirror
storcli /cx/dall show mirror
storcli /cx/dall add mirror src=<val>[force]
storcli /cx/dx set hidden=<on|off>
```

storcli /*cx*/dall show

This command shows the topology information of all the drive group.

Input example:

```
storcli /c0/dall show
```

storcli /*cx*/dall show all

This command shows all available configurations in the controller, which includes topology information, virtual drive information, physical drive information, free space, and free slot information.

Input example:

```
storcli /c0/dall show all
```

storcli /c*x*/dall show cachecade

This command shows all CacheCade virtual drive information.

Input example:

```
storcli /c0/dall show cachecade
```

storcli /c*x*/d*x* show

This command shows the topology information of the drive group.

Input example:

```
storcli /c0/d0 show
```

storcli /c*x*/d*x* show all

This command shows the physical drive and the virtual drive information for the drive group.

Input example:

```
storcli /c0/d0 show all
```

storcli /c*x*/d*x* set security=on[all]

This command enables security on the specified drive group.

Input example:

```
storcli /c0/d0 set security=on all
```

storcli /c*x*/d*x* split mirror

This command enables you to perform a break mirror operation on a drive group. The break mirror operation enables a RAID 1 configured drive group to be broken into two volumes. You can use one of the volumes in another system and replicate it without making a copy of the virtual drive.

> **NOTE**
> Break mirror is an offline operation. This command is supported only in UEFI.

Input example:

```
storcli /c0/d0 split mirror
```

storcli /c*x*/dall show mirror

This command shows information about the mirror associated with the drive group.

Input example:

```
storcli /c0/dall show mirror
```

storcli /c*x*/dall add mirror src=<*val*>[force]

This command joins the virtual drive with its mirror. The possible values to be used are 0, 1, or 2.

Input example:

```
storcli /c0/dall add mirror src=<1>[force]
```

storcli /c*x*/d*x* set hidden=<on|off>

This command hides or unhides a drive group.

Input example:

```
storcli /c0/d0 set hidden=on
```

Virtual Drive Power Settings Commands

Change Virtual Drive Power Settings Commands

The StorCLI utility supports the following commands to change the dimmer switch settings. You can use the following combinations for the Dimmer Switch commands:

```
storcli /cx set ds=off type=1|2|3|4
storcli /cx set ds=on type=1|2 [properties]
storcli /cx set ds=on type=3|4 defaultldtype=<value> [properties]
storcli /cx set ds=on [properties]
```

The following table describes the power-saving options.

Table 21: Dimmer Switch Input Options

Option	Value Range	Description
`dimmerswitch` or `ds`	`on`|`off`	Turns the dimmer switch option on.
`type`	`1` : Unconfigured `2` : Hot spare `3` : Virtual disk `4` : Unconfigured and hot spare drives	Specifies the type of drives for which the Dimmer Switch feature is applicable. **Note:** The dimmer switch is only activated for unconfigured drives and hot spare drives, but not for configured drives.
`properties`	`disableldps` : Interval in hours or time in *hh:mm* format `spinupdrivecount` : Valid enclosure number (0 to 255) `SpinUpEncDelay` : Valid time in seconds	Sets the interval or time in which the power-saving policy for the logical drive is turned off. Specifies the number of drives in the enclosure that are spun up. Specifies the delay of spin-up groups within an enclosure in seconds.

storcli /c*x* show DimmerSwitch (ds)

This command shows the current Dimmer Switch setting for the controller.

Input example:

```
storcli /c0 show ds
```

CacheVault Commands

The StorCLI Utility supports the following CacheVault commands:

```
storcli /cx/cv show
storcli /cx/cv show all
```

```
storcli /cx/cv show status
storcli /cx/cv start learn
```

The detailed description for each command follows.

storcli /cx/cv show

This command shows the summary information for the CacheVault of a controller.

Input example:

```
storcli /c0/cv show
```

storcli /cx/cv show all

This command shows all the information of the CacheVault.

> **NOTE**
>
> This command only works when a CacheVault is connected to the controller; otherwise, an error message appears. A capacitance value above 100% will be displayed as 100%.

Input example:

```
storcli /c0/cv show all
```

storcli /cx/cv show status

This command shows the battery information, firmware status, and the gas gauge status.

Input example:

```
storcli /c0/cv show status
```

storcli /cx/cv start learn

This command starts the CacheVault learning cycle. The battery learn cycle is immediately started and no other parameters are required for this command.

Input example:

```
storcli /c0/cv start learn
```

Enclosure Commands

The StorCLI utility supports the following enclosure commands:

```
storcli /cx/ex show
storcli /cx/ex show all
storcli /cx/ex download src=filepath [mode=5 | [forceActivate] mode=7] [bufferid=<val>]
storcli /cx/ex download src=filepath mode=e offline [forceActivate [delay=val]] [bufferid=<val>]
storcli /cx/ex download mode=f offline [delay=val] [bufferid=<val>]
storcli /cx/ex show status
storcli /cx/ex show phyerrorcounters
```

The detailed description for each command follows.

NOTE

StorCLI supports and can be used to qualify only Broadcom expanders and enclosures.

storcli /*cx*/*ex* show

This command shows the basic enclosure information.

Input example:

```
storcli /c0/e25 show
```

storcli /*cx*/*ex* show all

This command shows all enclosure information, which includes general enclosure information, enclosure inquiry data, a count of enclosure elements, and information about the enclosure elements.

Input example:

```
storcli /c0/e25 show all
```

storcli /*cx*/*ex* download src=*filepath* [mode=5 | [forceActivate] mode=7] [bufferid=<*val*>]

This command flashes the firmware with the file specified at the command line. The enclosure performs an error check after the operation. The following option can be used with the enclosure firmware download command.

Table 22: Enclosure Firmware Download Command Options

Option	Value Range	Description
forceactivate	—	Issues a command descriptor block (CDB) with write command with no data with command mode 0x0F (flash download already in progress). This option is used primarily to activate Scotch Valley enclosures.

NOTE
The firmware file that is used to flash the enclosure can be any format. The StorCLI utility assumes that you provide a valid firmware image.

Input example:

```
storcli /c0/e0 download src=c:\file2.bin
```

storcli /*cx*/*ex* download src=*filepath* mode=e offline [forceActivate [delay=*val*]] [bufferid=<*val*>]

This command supports the drive firmware using Mode E. Mode E downloads the microcode and allows you to issue this command for multiple devices.

NOTE
You can download as well as activate the drive firmware by executing the `activenow` command in the same command line. You can also specify the delay time, but the delay time specified by you is applicable only for activation and not for downloading the drive firmware.

Syntax:

```
storcli /cx/ex download src= mode=e offline [forceActivate]
```

Where:

- /c*x* – Specifies the controller, where *x* is the index of the controller.
- /e*x* – Specifies the enclosure ID of the controller (optional).

Input example:

```
storcli /c0/e25 download src=file.rom mode=E offline
```

storcli /c*x*/e*x* download mode=f offline [delay=*val*] [bufferid=<*val*>]

This command supports the drive firmware using Mode F. Mode F activates the deferred microcode and allows you to issue this command to all devices in a safe manner. You cannot issue this command before issuing the `Mode E` command. The default delay time is 15 seconds. You can specify any delay time between 1 and 300 seconds.

Syntax:

```
storcli /cx/ex download mode=f offline
```

Where:

- /c*x* – Specifies the controller, where *x* is the index of the controller.
- /e*x* – Specifies the enclosure ID of the controller (optional).

Input example:

```
storcli /c0/e25/ download mode=F offline delay=15
```

storcli /c*x*/e*x* show status

This command shows the enclosure status and the status of all enclosure elements.

Input example:

```
storcli /c0/e25 show status
```

storcli /c*x*/e*x* show phyerrorcounters

This command displays enclosure and expander phy error counters.

Syntax:

```
storcli /cx/ex show phyerrorcounters
```

Where:

- /c*x* – Specifies the controller, where *x* is the index of the controller.
- /e*x* – Specifies the enclosure, where *x* is the enclosure device ID.

Input example:

```
storcli /c0/e25 show phyerrorcounters
```

PHY Commands

The StorCLI Utility supports the following phy commands:

```
storcli /cx/px|pall set linkspeed=0(auto)|1.5|3|6|12
storcli /cx/px|pall show
storcli /cx/px|pall show all
storcli /cx/ex show phyerrorcounters
storcli /cx/ex/sx reset phyerrorcounters
```

The detailed description for each command follows.

storcli /c*x*/p*x*|pall set linkspeed=0(auto)|1.5|3|6|12

This command sets the PHY link speed. You can set the speed to 1.5Gb/s, 3Gb/s, 6Gb/s, or 12Gb/s. The link speed is set to `auto` when you specify `linkspeed = 0`.

Input example:

```
storcli /c0/p0 set linkspeed=1.5
```

storcli /c*x*/p*x*|pall show

This command shows the basic PHY layer information.

Input example:

```
storcli /c1/p0 show
```

storcli /c*x*/p*x*|pall show all

This command shows all the PHY layer information.

Input example:

```
storcli /c1/p0 show all
```

storcli /c*x*/e*x* show phyerrorcounters

This command shows the enclosure/expander phy error counters.

Input example:

```
storcli /c1/e0 show phyerrorcounters
```

storcli /c*x*/e*x*/s*x* reset phyerrorcounters

This command resets the drive phy error counters.

Input example:

```
storcli /c1/e0/s0 reset phyerrorcounters
```

PCIe Storage Interface Commands

The PCIe Storage Interface is the fundamental interface that connects peripheral devices to the host processor and through a memory controller to the memory architecture in the system. The PCIe interface communicates over one or more lanes that consist of one transmit and one receive serial interface for each lane.

Lane Speed Commands

The StorCLI Utility supports the following lane speed commands:

```
storcli /cx/lnx show
storcli /cx/lnall show
storcli /cx/lnx set lanespeed=0(disabled)|2.5|5|8|16
```

The detailed description for each command follows.

storcli /cx/lnx show

This command displays the lane information.

Input example:

```
storcli /c0/ln1 show
```

storcli /cx/lnall show

This command displays the summary information on all of the exiting lanes.

Input example:

```
storcli /c0/lnall show
```

storcli /cx/lnx set lanespeed=0 (disabled) | 2.5 |5 | 8 | 16

This command sets the lane speed. You can set the speed as `0(disabled)`, `2.5GT/s`, `5GT/s`, `8GT/s`, or `16GT/s`.

By default, the lane speed in the controller is 8GT/s or the value last saved by you.

Input example:

```
storcli /c0/ln1 set lanespeed=2.5
```

Output example:

Figure 1: Lane Speed Output

Link Configuration Commands

The StorCLI Utility supports the following link configuration commands:

```
storcli /cx/show linkconfig
storcli /cx/set linkconfig [connname=cx,cy] linkconfig=<val>
```

The detailed description for each command follows.

storcli cx/show linkconfig

This command displays the link configuration information for the current link configuration, pending link configuration, and the available link configuration.

Input example:

```
storcli /c1/show linkconfig
```

Output example: Current Link Configuration

Figure 2: Current Link Configuration

Output example – Pending Link Configuration

Figure 3: Pending Link Configuration

Output example: Available Link Configuration

Figure 4: Available Link Configuration

storcli /cx set linkconfig [connname=cx,cy] linkconfig=<*val*>

This command helps you configure the links for different ports of a controller.

Input example:

```
storcli /c1/set linkconfig connname=c0,c1 linkconfig=x4
```

Logging Commands

The StorCLI utility supports the following commands to generate and maintain log files:

```
storcli /cx delete events
storcli /cx delete termlog
storcli /cx show events file=<absolute path>
storcli /cx show eventloginfo
storcli /cx show termlog type=config|contents [logfile[=filename]]
storcli /cx show dequeuelog file =<filepath>
storcli /cx show alilog [logfile[=filename]]
```

The detailed description for each command follows.

storcli /cx delete events

This command deletes all records in the event log.

Input example:

```
storcli /c0 delete events
```

storcli /cx delete termlog

This command clears the TTY (firmware log for issue troubleshooting) logs.

Input example:

```
storcli /c0 delete termlog
```

storcli /cx show events file=<absolute path>

This command prints the system log to a text file and saves the file in the specified location.

Input example:

```
storcli /c0 show events file=C:\Users\brohan\test\eventreports
```

> **NOTE**
>
> The command output for this command cannot be JSON formatted.

storcli /cx show eventloginfo

This command shows the history of log files generated.

Input example:

```
storcli /c0 show eventloginfo type=config
```

> **NOTE**
>
> The command output for this command cannot be JSON formatted.

storcli /c*x* show termlog type=config|contents [logfile[=*filename*]]

This command shows the firmware logs. The `config` option shows the term log configuration (settings of TTY BBU buffering); the `contents` option shows the term log. The `contents` option is the default.

If you use the `logfile` option in the command syntax, the logs are written to the specified file. If you do not specify a file name, then the logs are written to the `storsas.log` file. If you do not use the `logfile` option in the command syntax, the entire log output is printed to the console.

Input example:

```
storcli /c0 show termlog=contents logfile = log.txt
```

> **NOTE**
>
> The command output for this command cannot be JSON formatted.

storcli /c*x* show dequeuelog =<filepath>

This command shows the debug log from the firmware.

Input example:

```
storcli /c0 show dequeuelog=<c:\test\log.txt>
```

> **NOTE**
>
> The command output for this command cannot be JSON formatted.

storcli /c*x*show alilog [logfile[=*filename*]]

This command gets the controller property, TTY logs, and events to the specified file.

Input example:

```
storcli /c0 show alilog logfile = log.txt
```

> **NOTE**
>
> The command output for this command cannot be JSON formatted.

Automated Physical Drive Configurations

The StorCLI utility supports the following automated physical drive configuration commands:

```
storcli /cx set autoconfig=r0 [immediate]
storcli /cx show autoconfig
storcli /cx set autoconfig=JBOD
storcli /cx set jbod=on|off
storcli /cx set autoconfig [=<none | R0 [immediate] | JBOD> [usecurrent] ] [[sesmgmt=on|off]  [securesed=on|
off] [multipath=on|off] [multiinit=on|off] [discardpinnedcache=Val>] [failPDOnReadME=on|off]  [Lowlatency=low|
off]]
```

The detailed description for each command follows.

storcli /c*x* set autoconfig=r0 [immediate]

This command lets you set the controller's automated configuration policy to RAID 0. When set to RAID 0, all unconfigured physical drives are configured as single RAID 0 drives, until the maximum virtual drive limit is reached. The `immediate` option lets this command execute the conversion (to RAID 0) operation only on all the existing

`Unconfigured Good` drives. Any physical drives newly connected in the future do not get converted to RAID 0. If you omit the `immediate` option in this command, all attached `Unconfigured Good` drives will become single drive RAID 0, and the `autoconfig` setting will become R0. Conversion to RAID 0 does not take place on newly connected physical drives, and the `autoconfig` setting will become R0. There is no setting that allows newly added `Unconfigured Drives` to be automatically converted to RAID 0. Automatic conversion to RAID 0 can be turned off by setting the `autoconfig policy` to `none`, which causes all drives to be converted to `Unconfigured Good`.

Input examples:

```
storcli /c0 set autoconfig=r0 immediate [[sesmgmt=[on|off]] [securesed=[on|off]] [multipath=[on|off]
[multiinit=[on|off]]]

storclif /c0 set autoconfig [=<none | R0 [immediate] | JBOD> [usecurrent] ] [[sesmgmt=on|off]  [securesed=on|
off] [multipath=on|off] [multiinit=on|off] [discardpinnedcache=<Val>] [failPDOnReadME=on|off]
 [Lowlatency=low|off]]
```

storcli /cx show autoconfig

This command lets you view the automatic physical drive policy.

Input example:

```
storcli /c0 show autoconfig
```

storcli /cx set autoconfig=JBOD

This command lets you set the controller's automated physical drive policy to `JBOD`. When set to `JBOD`, all unconfigured physical drives are configured as JBODs.

> **NOTE**
> If this command fails, enable the legacy JBOD mode first and retry the command.

Input example:

```
storcli /c0 set autoconfig=JBOD
```

storcli /cx set jbod=on|off

This command lets you enable the legacy JBOD mode.

Input example:

```
storcli /c0 set jbod=on
```

storcli /cx set autoconfig [=none | R0 [immediate] | JBOD> [usecurrent]] [[sesmgmt=on|off] [securesed=on|off] [multipath=on|off] [multiinit=on|off] [discardpinnedcache=Val>] [failPDOnReadME=on|off] [Lowlatency=low|off]]

This command enables or disables autoconfig on the selected adapter(s). You can use the following options with the `set autoconfig` command.

Table 23: Set Autoconfig Command Options

Option	Description
none	Disable autoconfig.
r0	Autoconfig for single PD as r0.

Option	Description
JBOD	Autoconfig each PD as JBOD.
immediate	Configure.
unconfigured	Pds once and will not configure any newly inserted UG drives and this option is only for R0.
usecurrent	Use current parameter values that are supported while changing the mode.
sesmgmt	Enable or disable SES management.
securesed	Enable or disable Security on SED.
multiinit	Enable or disable multi init.
multipath	Enable or disable multi path.

Input example:

```
storclif /c0 set autoconfig [=R0 [immediate]> [usecurrent] ] [[sesmgmt=on]  [securesed=on] [multipath=on]
  [multiinit=on] [discardpinnedcache=Val>] [failPDOnReadME=on]  [Lowlatency=low]]
```

Recovery Commands (UEFI Only)

Recovery commands perform recovery actions related to a specified controller. Recovery commands are supported on UEFI environment only. The Storage Command Line Interface Tool supports the following recovery commands.

```
storcli /cx download completeflash fileone=<IT boot loader image> filetwo=<firmware image>
```

```
storcli /cx erase all [excludemfg]
```

```
erase all [excludemfg]
```

The detailed description of each command follows:

storclio /*cx* download completeflash fileone=<*IT boot loader image*> filetwo=<*firmware image*>

This command downloads the complete flash image on a non-operational or an empty controller by performing host boot using the IT boot loader image. This command takes two files as arguments:

- `Fileone` — A valid Itboot loader image with which host boot is performed on the controller.
- `Filetwo` — A valid firmware image which is flashed on the controller.

Syntax:

```
storcli /c1 download completeflash fileone=<Itbootloaderimage> filetwo=<FW image>
```

Where:

`/cx` — Specifies the controller where `x` is the index of the controller, and `filenames` are the arguments.

Input examples:

```
storcli /c1 download completeflash fileone=vtboot.rom filetwo=nopad.rom
```

> **NOTE**
>
> Unified StorCLI can flash only NoPad image. It cannot flash 16 MB/32 MB images.

storcli /*cx* erase all [excludemfg] file=<*itbootloader image*>

This command erases the complete flash region, but retains the manufacturing data region.

Syntax:

```
storcli /cx erase all [excludemfg] file=<itbootloader image>
```

Input examples:

```
storcli /c1 erase all excludemfg file=vtboot.rom
```

> **NOTE**
>
> The StorCLI tool supports only the `erase all excludemfg` **erase option. It does not support the** `erase all` option.

Frequently Used Tasks

Displaying the Version of the StorCLI Utility

The following command displays the version of the command line tool:

```
storcli -v
```

Displaying the StorCLI Tool Help

The following command displays the StorCLI utility help:

```
storcli -h
```

Help appears for all the StorCLI tool commands.

Displaying System Summary Information

The following command displays the summary of all the controller information:

```
storcli -show [all]
```

Displaying Free Space in a Controller

The following command displays the free space available in the controller:

```
storcli /cx show freespace
```

Adding Virtual Drives

The following command creates a virtual drive:

```
storcli /cx add vd type=raid[0|1|5|6|10|50|60][Size=<VD1_Sz>,<VD2_Sz>,..|*all]
[name=<VDNAME1>,..] drives=e:s|e:s-x|e:s-x,y [PDperArray=x|auto*]
[SED] [pdcache=on|off|*default][pi] [DimmerSwitch(ds)=default|automatic(auto)|
*none|maximum(max)|MaximumWithoutCaching(maxnocache)] [wt|*wb|awb] [nora|*ra]
[*direct|cached]
[strip=<8|16|32|64|128|256|512|1024] [AfterVd=x] [Spares=[e:]s|[e:]s-x|[e:]s-x,y]
```

> **NOTE**
>
> The supported strip size can vary from a minimum of 64 KB to 1 MB for MegaRAID controllers and only 64 KB for Integrated MegaRAID controllers.

```
[Cbsize = 0|1|2 Cbmode = 0|1|2]
[force]
```

The following inputs can be used when adding virtual drives:

- The controller in which the virtual drives are created.
- The RAID type of the virtual drives.
  The supported RAID types are 0, 1, 5, 6, 10, 50, and 60.
- The size of each virtual drive.
- The drives that create the virtual drives.
  Drives = $e:s|e:s-x|e:s-x,y$
  Where:
  - e specifies the enclosure ID.
  - s represents the slot in the enclosure.
  - $e:s-ex$ is the range conventions used to represents slots s to x in the enclosure e.
- The physical drives per array.
  The physical drives per array can be set to a particular value.
- The SED option creates security-enabled drives.
- The PDcache option can be set to on or off .
- The pi option enables protection information.
- The dimmerswitch is the power save policy. It can be set to default or automatic * , none , maximum(max) , or MaximumWithoutCaching(maxnocache) .
- The wt option disables write back.
- The nora option disables read ahead.
- The cached option enables the cached memory.
- The strip option sets the strip size.
  It can take the values 8 , 16 , 32 , 64 , 128 , 256 , 512 , or 1024 .

  > **NOTE**
  > The supported strip size can vary from a minimum of 64 KB to 1 MB for MegaRAID controllers, and only 64 KB for Integrated MegaRAID controllers.

- The AfterVdX option creates the virtual drives in the adjacent free slot next to the specified virtual drives.

  > **NOTE**
  > The * indicates default values used in the creation of the virtual drives. If values are not specified, the default values are taken.

Input example:

This command creates a RAID volume of RAID 1 type from drives in slots 10 to slot 15 in enclosure 0. The strip size is 64 KB.

Setting the Cache Policy in a Virtual Drive

The following command sets the write cache policy of the virtual drive:

```
storcli /cx/v(x|all) set wrcache=wt|wb|awb
```

The command sets the write cache to write back, write through, or always write back.

Displaying Virtual Drive Information

The following command displays the virtual drive information for all the virtual drives in the controller:

```
storcli /cx/v(x/all) show
```

Deleting Virtual Drives

The following command deletes virtual drives:

```
storcli /cx/v(x|all) del [cc|cachecade]
```

The following inputs are required when deleting a virtual drive:

- The controller on which the virtual drive or virtual drives is present.
- The virtual drives that must be deleted; or you can delete all the virtual drives on the controller using the `vall` option.
- The `cc` or `cachecade` option to confirm that the deleted drive is a CacheCade drive.

Flashing Controller Firmware

The following command is used to flash the controller firmware.

```
storcli /cx download file=filepath
[fwtype=<value>][nosigchk][noverchk][resetnow]
```

> **NOTE**
>
> The command output for this command cannot be JSON formatted.

Recovery Commands (UEFI Only)

Recovery commands perform recovery actions related to a specified controller. Recovery commands are supported on UEFI environment only.

The following commands are used:

```
storcli /cx erase all excludemfg file=<it bootloader image>
```

```
storcli /c1 erase all excludemfg file=vtboot.rom
```

Supported Commands on Initiator-Target Controllers

This chapter provides the information about StorCLI for Initiator-Target (IT) controllers (HBAs).

System Commands

System commands gather information about the controller, such as drive topology, number of controllers connected, controller properties, controller-associated information, and so on.

The StorCLI tool supports the following system commands.

```
storcli show
storcli show all
storcli show ctrlcount
storcli show help
storcli v
storcli help
storcli help IT
```

The detailed description for each command follows:

storcli /cx show

This command shows the number of drives connected to a specific controller.

Syntax:

```
storcli /cx show
storcli /call show
```

Where:

- `cx` : Specifies the controller where `x` is the index of the controller.
- `all` : Displays information on all the controllers present on the host.

Input examples:

```
storcli /c0 show
storcli /call show
```

storcli /cx show all

This command shows the list of controllers and its associated information. It also displays information about the drives that need attention.

Syntax:

```
storcli /cx show all
storcli /call show all
```

Where:

- `c` : Specifies the controller where `x` is the index of the controller.
- `all` : Displays information on all the controllers present on the host.

Input examples:

```
storcli /c0 show
storcli /call show all
```

storcli show ctrlcount

This command lists the number of controllers that are connected to the system.

Input example:

```
storcli show ctrlcount
```

storcli show help

This command shows help for all commands that has the keyword `show`.

Input example:

```
storcli show help
```

storcli v

This command shows the version of the StorCLI tool.

Input example:

```
storcli -v
```

storcli help

This command shows help for all commands at the server level.

Input example:

```
storcli help
```

storcli help IT

This command shows help for all the StorCLI IT commands.

Input example:

```
storcli help IT
```

SAS Address Commands

SAS address commands gather SAS address information, such as displaying the SAS address of a specific controller. These commands also allow you to specify a new SAS address to the controller.

StorCLI supports the following SAS address commands:

```
storcli /cx show sasadd
storcli /cx set sasadd=<sasaddress>
storcli /cx set sasaddhi=<value> sasaddlow=<value>
```

The detailed description for each command follows.

storcli /*cx* show sasadd

This command displays the SAS address of the specified controller.

Syntax:

```
storcli /cx show sasadd
```

Where:

/c*x* – Specifies the controller where *x* is the index of the controller.

Usage example:

```
storcli /c0 show sasadd
```

storcli /*cx* set sasadd=*<value>*

Using this command, you can change the existing SAS address on a specific controller with a SAS address of your choice.

> **NOTE**
> Make sure you provide the complete SAS address in a hexadecimal format.

Syntax:

```
storcli /cx set sasadd=<xxxxxxxxxxxxxxx>
```

Where:

/c*x* – Specifies the controller where *x* is the index of the controller.

Usage example:

```
storcli /c0 set sasadd=50062b000000000
```

storcli /*cx* set sasaddhi=*<value>* sasaddlow=*<value>*

Using this command, you can program the SAS Address High and SAS Address Low to the specified controller. Using the `set sasaddhi` command, you can program the specified controller with the first 28 bits (seven characters) of the SAS address, which are specified at the command line. Using the `set sasaddlow`, the other 36 bits (nine characters) are programmed to the controller when the controller prompts you to enter the value.

Usage example:

```
storcli /c0 set sasaddhi=5062b00 sasaddlow=000000000
```

storcli /*cx* set sasadd =*<sas address>* devicename

Using this command, you can program a specific SAS address to a specified controller. Also, you can program the device name for each phy by providing the SAS address as input.

Usage example:

```
storcli /c0 set sasadd=50062b000000000
```

storcli /*cx* set sasadd=*<sas address>* methodport

Using this command, you can program the SAS address to a specific controller. You can also program each physical phy in the I/O unit with a SAS address.

Usage example:

```
storcli /c0 set sasadd=50062b000000000 methodport
```

storcli /*cx* set sasaddhi=*<value>* sasaddlow=*<value>* methodport

Using this command, you can program a SAS address to a specific controller. You can also program each physical SAS phy in the I/O unit with a SAS address.

Usage example:

```
storcli /cx set sasaddhi=5062b00 sasaddlow = 000000000 methodport
```

Product Settings and Display Commands

Product settings and display commands are used to gather information about the product, such as assembly value and tracer value of a specific controller.

StorCLI supports the following SAS address commands:

```
storcli /cx show AssemblyNumber
storcli /cx set AssemblyNumber=<assemblynumber>
storcli /cx show TracerNumber
storcli /cx set TracerNumber=<tracernumber>
storcli /cx show vpd
storcli /cx set updatevpd file=<filepath>
```

storcli /*cx* show AssemblyNumber

This command displays the assembly value of the specified controller.

Syntax:

```
storcli /cx show AssemblyNumber
```

Where:

/c*x* – Specifies the controller where *x* is the index of the controller.

Input example:

```
storcli /c1 show AssemblyNumber
```

storcli /*cx* set AssemblyNumber

Using this command, you can program the specified controller with an assembly value. The assembly value can take up to 16 characters, which is programmed directly to the controller. The value is truncated or padded with termination characters, as required.

Syntax:

```
storcli /cx set assemblynumber=<value>
```

Where:

/c*x* — Specifies the controller where *x* is the index of the controller, and *assemblynumber* is the value of the assembly that you need to specify.

Input example:

```
storcli /c1 set AssemblyNumber=12AB
```

storcli /*cx* show TracerNumber

This command displays the tracer value of the specified controller.

Syntax:

```
storcli /cx show TracerNumber
```

Where:

/*cx* — Specifies the controller where *x* is the index of the controller.

Input example:

```
storcli /c0 show TracerNumber
```

storcli /*cx* set TracerNumber=<*TracerNumber*>

Using this command, you can program the specified controller with a board tracer value. The assembly value can take up to 16 characters. When you enter the 16-character value on the command line, the StorCLI utility programs it to the controller. The board tracer value is truncated or padded with termination characters, as needed.

Syntax:

```
storcli /cx set TracerNumber=<tracernumber>
```

Where:

/*cx* — Specifies the controller where *x* is the index of the controller, and *tracernumber* is the board tracer value that you specify.

Input example:

```
storcli /c0 set TracerNumber=12AB
```

storcli /*cx* show vpd

This command displays the contents of the Vital Product Data (VPD) for a specified controller.

Syntax:

```
storcli /cx show vpd
```

Where:

/*cx* — Specifies the controller where *x* is the index of the controller.

Input example:

```
storcli /c0 show vpd
```

> **NOTE**
>
> The command output for this command cannot be JSON formatted.

storcli /*cx* set updatevpd file=<*filepath*>

Using this command, you can program the VPD file for a specified controller. The StorCLI utility parses the data to the file that you specify on command line and programs the parsed data to the appropriate location in the nonvolatile memory. StorCLI verifies the individual field lengths, but it does not verify the parameter value.

Syntax:

```
storcli /cx set updatevpd file=<filepath>
```

Where:

/c*x* — Specifies the controller where *x* is the index of the controller, and *filepath* is the location of the file on which you want to parse the data.

Input example:

```
storcli /c0 set updatevpd file=c:\vpd\vpd.ini
```

> **NOTE**
>
> The command output for this command cannot be JSON formatted.

Upgrade, Downgrade, and Recovery Commands

These commands are used to upgrade, downgrade, and recover the firmware.

StorCLI supports the following upgrade, downgrade, and recovery commands. The `download completeflash` and `erase` commands are applicable only in the UEFI environment.

```
storcli /cx download file=<filepath>
storcli /cx download bios file=<filepath>
storcli /cx download efibios file=<filepath>
storcli /cx download fcode file=<filepath>
storcli /cx download cpld file=<filepath>
storcli /cx download completeflash fileone=<firmwareimage> filetwo=<flashimage>
storcli /cx erase nvsram
storcli /cx erase fwbackup
storcli /cx erase perconfpage
storcli /cx erase mpb
storcli /cx erase bootservices
storcli /cx erase all
storcli /c0 erase all excludemfg
storcli /cx download psoc file=<filepath>
```

storcli /c*x* download file=<*filepath*>

Using this command, you can flash the firmware with the `.rom` file to a specified adapter from the provided file location.

Syntax:

```
storcli /cx download file=<filepath>
```

Where:

/c*x* – Specifies the controller, where *x* is the index of the controller, and *filepath* specifies the absolute file path.

Input example:

```
storcli /c0 download file=image.fw
```

> **NOTE**
> Downgrading the firmware between phases is not supported on Initiator-Target controllers.

Download Commands

storcli /*cx* download bios file=<*.rom*>

Using this command, you can update the BIOS component on all supported controllers.

Syntax:

```
storcli /cx download bios file=<.rom>
```

Where:

/c*x* — Specifies the controller, where *x* is the index of the controller, and .rom specifies the file extension on which you are updating the BIOS component.

Input example:

```
storcli /c0 download bios file=mptsas.rom
```

storcli /*cx* download fcode file=<*.rom*>

Using this command, you can update the FCODE component on all supported controllers.

Syntax:

```
storcli /cx download fcode file=<.rom>
```

Where:

/c*x* — Specifies the controller, where *x* is the index of the controller, and .rom specifies the file extension on which you are updating the FCODE component.

Input example:

```
storcli /c0 download fcode file=mptsas.rom
```

storcli /*cx* download cpld file=<*.rom*>

Using this command, you can update the CPLD component on all supported controllers.

Syntax:

```
storcli /cx download cpld file=<.rom>
```

Where:

/c*x* — Specifies the controller, where *x* is the index of the controller, and .rom specifies the file extension on which you are updating the CPLD component.

Input example:

```
storcli /c0 download cpld file=cpld.rom
```

UEFI Commands

storcli /*cx* download completeflash

This command downloads the complete flash image on a nonoperational or an empty controller by performing host boot using the IT boot loader image. This command takes two files as arguments:

- fileone — A valid firmware image with which host boot is performed on the controller.
- filetwo — A valid flash image that is flashed on the controller.

Syntax:

```
storcli /cx download completeflash fileone=<firmwareimage> filetwo=<flashimage>
```

Where:

`/cx` — Specifies the controller, where `x` is the index of the controller, and `filenames` are the arguments.

The firmware image for your controller and current firmware release version might be available on the Broadcom website. Contact your local FAE representative for support and clarification of which firmware release is available for your specific IT controller.

Input example:

```
storcli /c0 download completeflash fileone=vtfirm3616.rom filetwo=FWimage.rom
```

> **NOTE**
> The `fileone=vtfirm3616.rom` option is provided by the FAE representative, which supports only the model SAS3616 IT controller.

storcli /cx erase nvsram

Using this command, you can clear or erase the NVSRAM region.

> ⚠️ **CAUTION**
> Use this command and all other `erase` commands with caution. The `erase` commands can clear or erase any previous configuration present on the controller, and you cannot undo any erase operation.

Syntax:

```
storcli /cx erase nvsram
```

Where:

`/cx` — Specifies the controller, where `x` is the index of the controller.

Input example:

```
storcli /c0 erase nvsram
```

storcli /cx erase fwbackup

Using this command, you can clear or erase the firmware backup region.

Syntax:

```
storcli /cx erase fwbackup
```

Where:

`/cx` — Specifies the controller, where `x` is the index of the controller.

Input example:

```
storcli /c0 erase fwbackup
```

storcli /cx erase perconfpage

Using this command, you can clear or erase the Persistent Configuration page.

Syntax:

```
storcli /cx erase perconfpage
```

Where:

/c*x* — Specifies the controller, where *x* is the index of the controller.

Input example:

```
storcli /c0 erase perconfpage
```

storcli /c*x* erase mpb

Using this command, you can clear or erase the manufacturing area details.

Syntax:

```
storcli /cx erase mpb
```

Where:

/c*x* — Specifies the controller, where *x* is the index of the controller.

Input example:

```
storcli /c0 erase mpb
```

storcli /c*x* erase bootservices

Using this command, you can clear or erase the boot services region.

Syntax:

```
storcli /cx erase bootservices
```

Where:

/c*x* — Specifies the controller, where *x* is the index of the controller.

Input example:

```
storcli /c0 erase bootservices
```

storcli /c*x* erase all [excludemfg]

Using this command, you can clear or erase the complete flash region. If you use the `excludemfg` option, this command erases the flash region but retains the manufacturing data region.

Syntax:

```
storcli /cx erase all excludemfg
```

Where:

/c*x* — Specifies the controller, where *x* is the index of the controller, and specifying `excludemfg` as an option erases the flash region except the manufacturing data region.

Input example:

```
storcli /c0 erase all excludemfg
```

storcli /c*x* erase all

Using this command, you can clear or erase the complete flash region.

Syntax:

```
storcli /cx erase all
```

Where:

/c*x* — Specifies the controller, where *x* is the index of the controller, and specifying the erase all option erases the flash region.

Input example:

```
storcli /c0 erase all
```

Drive Commands

Drive commands are used to start and stop locating physical drives.

StorCLI supports the following drive commands:

```
storcli /cx/ex/sx start locate
storcli /cx/ex/sx stop locate
storcli /cx show bootdrive
storcli /cx[/ex]/sx set bootdrive=[on|off]
storcli /cx/ex/sx download src=<filepath> [mode=5|7]
storcli /cx[/ex]/sx download src=<<filename>> mode=E [activatenow [delay=<val>]]
storcli /cx[/ex]/sx download mode=f [delay=<val>]
```

storcli /c*x*/e*x*/s*x* start locate

Using this command, you can turn on the drive LED flash to locate physical drives.

Syntax:

```
storcli /cx/ex/sx start locate
```

Where:

/c*x* – Specifies the controller, where *x* is the index of the controller.

/e*x* – Specifies the enclosure ID of the controller, which is optional.

/s*x* – Refers to the drive slot ID of the controller.

Input example:

```
storcli /c0/e25/s4 start locate
```

storcli /c*x*/e*x*/s*x* stop locate

Using this command, you can turn off the drive LED flash to locate physical drives.

Syntax:

```
storcli /cx/ex/sx stop locate
```

Where:

/c*x* – Specifies the controller, where *x* is the index of the controller.

/e*x* – Specifies the enclosure ID of the controller, which is optional.

/s*x* – Specifies the drive slot ID of the controller.

Input example:

```
storcli /c0/e25/s4 stop locate
```

storcli /c*x* show bootdrive

Use this command to show the boot drive configured on the controller.

Syntax:

```
storcli /cx show bootdrive
```

Where:

`/cx` – Specifies the controller, where *x* is the index of the controller.

Input example:

```
storcli /c0 show bootdrive
```

> **NOTE**
> StorCLI displays only the primary boot device.

storcli /c*x*[/e*x*]/s*x* set bootdrive=[on|off]

Use this command to set or unset a physical drive as a boot drive.

Syntax:

```
storcli /cx[/ex]/sx set bootdrive=[on|off]
```

Where:

`/cx` – Specifies the controller, where *x* is the index of the controller.

`/ex` – Specifies the enclosure ID of the controller, which is optional.

`/sx` – Specifies the drive slot ID of the controller.

Input example:

```
storcli /c0/e25/s4 set bootdrive=on
```

> **NOTE**
> You can specify only the primary boot device.

storcli /c*x*/e*x*/s*x* download src=<*filename*> [mode=5|7]

Use this command to flash the drive firmware with the specified file.

The `mode` options specify the SCSI write buffer mode.

`5` – The entire drive firmware file is downloaded at once.

`7` – The drive firmware is downloaded in chunks of 32 KB.

> **NOTE**
> The default mode is `7` .

Syntax:

```
storcli /cx/ex/sx download src=filename [mode=5|7]
```

Where:

`/cx` – Specifies the controller, where *x* is the index of the controller.

`/ex` – Specifies the enclosure ID of the controller (which is optional).

`/sx` – Specifies the drive slot ID of the controller.

Input example:

```
storcli /c0/e25/s4 download src=c:\file1.bin mode=5
```

storcli /c*x*/e*x*/s*x* download src=<*filepath*> mode=E [activatenow [delay=<*val*>]]

This command supports the drive firmware using `Mode E` .

`Mode E` downloads the microcode and allows you to issue this command for multiple devices. You can use this command only in an offline mode.

Syntax:

```
storcli /cx/ex/sx download src=<filepath> mode=E [activatenow [delay=<val>]]
```

Where:

- `/cx` – Specifies the controller, where *x* is the index of the controller.
- `/ex` – Specifies the enclosure ID of the controller, which is optional.
- `/sx` – Specifies the drive slot ID of the controller.
- `delay` – The default delay is 15 seconds. The user can enter a delay from 1 to 300 seconds. The delay is only for the activation and not for the download.

Input example:

```
storcli /c0/e25/s4 download src=file.rom mode=E offline
```

storcli /c*x*[/e*x*]/s*x* download mode=F [delay=<*val*>]

This command supports the drive firmware using `Mode F` .

`Mode F` activates the deferred microcode and allows you to issue this command to all devices in a safe manner. You can use this command only in an offline mode. You cannot issue this command before issuing the Mode E command. The default delay time is 15 seconds. You can specify any delay time between 1 and 300 seconds.

> **NOTE**
>
> You can download and activate the drive firmware by issuing the `activenow` command in the same command line. You can also specify the delay time, but the delay time specified by you is applicable only for activation and not for downloading the drive firmware.

Syntax:

```
storcli /cx/ex/sx download mode=f [delay=<val>]
```

Where:

`/cx` – Specifies the controller, where *x* is the index of the controller.

`/ex` – Specifies the enclosure ID of the controller, which is optional.

`/sx` – Specifies the drive slot ID of the controller.

Input example:

```
storcli /c0/e25/sall download mode=F offline delay=15
```

Enclosure Commands

The StorCLI utility supports the following enclosure commands:

```
storcli /cx/ex show

storcli /cx/ex show all

storcli /cx/ex download src=<filepath> [mode=5 | [forceActivate] mode=7] [bufferid=<val>]

storcli /cx/ex download src=<filepath> mode=e [forceActivate [delay=val]] [bufferid=<val>]

storcli /cx/ex download mode=f [delay=val] [bufferid=<val>]
```

The detailed description for each command follows.

> **NOTE**
>
> StorCLI supports and can be used to qualify only Broadcom expanders and enclosures.

storcli /cx/ex show

This command shows basic enclosure information.

Input example:

```
storcli /c0/e25 show
```

storcli /cx/ex show all

This command shows all enclosure information, which includes general enclosure information, enclosure inquiry data, a count of enclosure elements, and information about the enclosure elements.

Input example:

```
storcli /c0/e25 show all
```

storcli /cx/ex download src=filepath [mode=5 | [forceActivate] mode=7] [bufferid=<val>]

This command flashes the firmware with the file specified at the command line. The enclosure performs an error check after the operation. The following option can be used with the enclosure firmware download command.

Table 24: Enclosure Firmware Download Command Options

Option	Value Range	Description
forceactivate	—	Issues a command descriptor block (CDB) with write command with no data with command mode 0x0F (flash download already in progress). This option is used primarily to activate Scotch Valley enclosures.

> **NOTE**
> The firmware file that is used to flash the enclosure can be any format. The StorCLI utility assumes that you provide a valid firmware image.

Input example:

```
storcli /c0/e0 download src=c:\file2.bin
```

storcli /cx/ex download src=filepath mode=e offline [forceActivate [delay=val]] [bufferid=<val>]

This command supports the drive firmware using Mode E. Mode E downloads the microcode and allows you to issue this command for multiple devices.

NOTE
You can download and activate the drive firmware by issuing the `activenow` command in the same command line. You can also specify the delay time, but the delay time specified by you is applicable only for activation and not for downloading the drive firmware.

Syntax:

```
storcli /cx/ex download src=<filepath> mode=e [forceActivate]
```

Where:

- `/cx` – Specifies the controller, where `x` is the index of the controller.
- `/ex` – Specifies the enclosure ID of the controller (optional).

Input example:

```
storcli /c0/e25 download src=file.rom mode=E
```

storcli /cx/ex download mode=f [delay=val] [bufferid=<val>]

This command supports the drive firmware using Mode F. Mode F activates the deferred microcode and allows you to issue this command to all devices in a safe manner. You cannot issue this command before issuing the `Mode E` command. The default delay time is 15 seconds. You can specify any delay time between 1 and 300 seconds.

Syntax:

```
storcli /cx/ex download mode=f
```

Where:

- `/cx` – Specifies the controller, where `x` is the index of the controller.
- `/ex` – Specifies the enclosure ID of the controller (optional).

Input example:

```
storcli /c0/e25/ download mode=F delay=15
```

Add RAID 0, 1, 5, or 6 Configuration

Use the command in the following table to add a RAID level 0, 1, 5, or 6 configuration to the existing configuration on the selected controller.

Table 25: Add RAID 0, 1, 5, or 6 Configuration

Convention	MegaCli -CfgLDAdd -R0|-R1|-R5|-R6[E0:S0,E1:S1,...] [WT | WB] [NORA | RA] [Direct | Cached] [CachedBadBBU|NoCachedBadBBU] [-szXXXXXXXX [-szYYYYYYYY [...]]] [-strpszM] [-Hsp[E5:S5,...]] [-afterLdX] [-Force] [-cache] [-enblPI]-aN
Description	Adds a RAID level 0, 1, 5, or 6 configuration to a specified controller. Even if no configuration is present, you have the option to write the configuration to the controller.
	RAID 1 supports up to 32 drives in a single span of 16 drive groups. RAID 1 requires an even number of drives, because data from one drive is mirrored to the other drive in each RAID 1 drive group.
	-Rx[E0:S0,...] : Specifies the RAID level and the drive enclosure/slot numbers to construct a drive group.
	-WT (Write Through), WB (Write Back): Selects the write policy.
	-NORA (No Read Ahead), RA (Read Ahead): Selects the read policy.

`-Cached`,`-Direct` : Selects the cache policy.

`[{CachedBadBBU|NoCachedBadBBU }]` : Specifies whether to use write cache when the BBU is bad.

`-szXXXXXXXX` : Specifies the capacity for the virtual drive, where `XXXX` is a decimal number of MB. However, the actual capacity of the virtual drive can be smaller, because the driver requires the number of blocks from the drives in each virtual drive to be aligned to the stripe size. If multiple size options are specified, CT configures the virtual drives in the order of the options entered in the command line. The configuration of a particular virtual drive fails if the remaining capacity of the drive group is too small to configure the virtual drive with the specified capacity. This option can also be used to create a configuration on the free space available in the drive group.

`-strpszM` : Specifies the stripe size, where the stripe size values are 8 KB, 16 KB, 32 KB, 64 KB, 128 KB, 256 KB, 512 KB, or 1024 KB.

`Hsp[E5:S5,...]` : Creates hot spares when you create the configuration. The new hot spares are dedicated to the virtual drive used in creating the configuration. This option does not let you create global hot spares. To create global hot spares, you must use the `-PdHsp` command with proper sub commands.

You can also use this option to create a configuration on the free space available in the virtual drive. You can specify which free slot must be used by specifying the `#afterLdX` . This command is optional. By default, the application uses the first free slot available in the virtual drive. This option is valid only if the virtual drive is already used for configuration.

`-enblPI` Indicates whether protection information is enabled on the controller or not. Values: 0: Disables protection information on the controller 1: Enables protection information on the controller.

Examples of Adding RAID 0, 1, 5, or 6 Configuration

Following are examples of some of the properties of the Add RAID 0, 1, 5, or 6 configuration command.

- `MegaCLI cfgldadd R0[21:1] a0`
  Adds RAID level 0 configuration to the selected drive (21:1).

- `MegaCLI cfgldadd R1[21:3,21:4] a0`
  Adds RAID level 1 configuration to the selected drives (21:3, 21: 4). RAID 1 requires an even number of drives.

- `MegaCLI cfgldadd R0 [21:1] Direct NORA WT strpsz8 a0`
  Adds a RAID level 0 configuration to the selected drive (21:1) with a Direct cache policy, a NORA read policy, a Write Through write policy, and a stripe size of 8 KB.

Compare Commands

The StorCLI supports the following controller commands:

`storcli /cx compare bios ver=<version number>`

`storcli /cx compare firmware ver=<version number>`

`storcli /cx compare fwprodid ver=<version number>`

`storcli /cx compare ssid ver=<version number>`

`storcli /cx/px compare linkspeed=<value>`

storcli /cx compare bios ver= <version number>

Using this command, you can compare the current BIOS version against a BIOS version provided in the command line. You must enter the BIOS version in the following format:

```
AA.BB.CCC.DD
```

Also, the StorCLI utility checks all four parts of the BIOS version number (`AA.BB.CCC.DD`), and displays the result of the checks through the return codes.

Syntax:

```
storcli /cx compare bios ver=<version number>
```

Where:

`/cx` – Specifies the controller, where `x` is the index of the controller, and `version number` specifies the BIOS version number that you want to compare.

Input example:

```
storcli /c0 compare bios ver=6.08.00.00
```

storcli /cx compare firmware ver=<version number>

Using this command, you can compare the current firmware version against a firmware version provided in the command line. You must enter the firmware version in the following format:
```
AA.BB.CC.DD
```
Also, the StorCLI utility checks all four parts of the firmware version number (`AA.BB.CC.DD`), and displays the result of the checks through the return codes.

Syntax:

```
storcli /cx compare firmware ver=<version number>
```

Where:

`/cx` – Specifies the controller, where `x` is the index of the controller, and `version number` specifies the firmware version number that you want to compare.

Input example:

```
storcli /c0 compare firmware ver=00.07.00.23
```

storcli /cx compare fwprodid ver=<version number>

Using this command, you can compare the current product ID of the current firmware version against the product ID provided in the command line. You must enter the product ID in following hexadecimal format:
```
AA.BB
```

Syntax:

```
storcli /cx compare fwprodid ver=<version number>
```

Where:

`/cx` — Specifies the controller where `x` is the index of the controller, and `version number` specifies firmware product ID that you want to compare.

Input example:

```
storcli /c0 compare fwprodid ver=09.33
```

storcli /cx compare ssid ver=<version number>

Using this command, you can test the Subsystem Vendor ID (SSVID) against the Subsystem Device ID (SSDID) to match them to the appropriate PCI information for a selected controller. The StorCLI displays in discrepancies, if found, in either the SSVID or the SSDID; otherwise, it displays a success message if both the values match the specified controller.

Syntax:

```
storcli /cx compare ssid ver=<version number>
```

Where:

`/cx` – Specifies the controller, where `x` is the index of the controller, and `version number` specifies the SSID number against which you want to compare the SSVID number.

Input example:

```
storcli /c0 compare ssid ver=1234:4567
```

storcli /c*x*/p*x* compare linkspeed= *<value>*

Using this command, you can test the link state of a specific phy on a specific controller against a provided value. The StorCLI utility displays the results through return codes.

Syntax:

```
storcli /cx/px compare linkspeed=<value>
```

Where:

`/cx` – Specifies the controller, where `x` is the index of the controller, `/px` specifies the PHY number, and `value` specifies the link status value that you provide.

The valid link status values for testing the link state are shown in the following table.

Table 26: Valid Link Status Values

Value	Link State
0	Link Down
1	3.0Gb/s
2	6.0Gb/s
3	12.0Gb/s

Input example:

```
storcli /c0/p0 compare linkspeed=2
```

Get Commands

The StorCLI utility supports the following `get` commands:

```
storcli cx get bios file=<filepath>

storcli cx get firmware file=<filepath>

storcli cx get fwbackup file=<filepath>

storcli cx get nvdata file=<filepath>

storcli cx get flash file=<filepath>
```

storcli /c*x* get bios file= *<filepath>*

Using this command, you can upload the BIOS block to a specified file. If the file already exists, this command overwrites the existing file. This command fails if there is not enough disk space to hold the BIOS block. This command is useful in manufacturing environments to perform a BIOS comparison after an update.

Syntax:

```
storcli /cx get bios file=<filepath>
```

Where:

/c*x* – Specifies the controller, where *x* is the index of the controller, and *filepath* refers to the location of the BIOS file.

Input example:

```
 storcli /c0 get bios file=image.rom
```

storcli /c*x* get firmware file= *<filepath>*

Using this command, you can upload the current firmware image to a specified file. If the file already exists, this command overwrites the existing file. This command fails if there is not enough disk space to hold the firmware image. This command is useful in manufacturing environments to perform a comparison after an update.

Syntax:

```
storcli /cx get firmware file=<filepath>
```

Where:

/c*x* – Specifies the controller, where *x* is the index of the controller, and *filepath* refers to the location of the BIOS file.

Input example:

```
 storcli /c0 firmware file=c:\fw\image.rom
```

storcli /c*x* get fwbackup file= *<filepath>*

Using this command, you can upload the backup image of the firmware to a file. If the file already exists, this command overwrites the existing file. This command fails if there is not enough disk space to hold the backup image of the firmware. This command is useful in manufacturing environments to perform a firmware image comparison after an update.

Syntax:

```
storcli /cx get fwbackup file=<filepath>
```

Where:

/c*x* – Specifies the controller, where *x* is the index of the controller, and *filepath* refers to the location of the backup image of the firmware.

Input example:

```
 storcli /c0 get fwbackup file=c:\fw\image.rom
```

storcli /c*x* get nvdata file= *<filepath>*

Using this command, you can upload the current binary NVDATA image to a file. If the file already exists, this command overwrites the existing file. This command fails if there is not enough disk space to hold the binary NVDATA image. This command is useful in manufacturing environments to perform an NVDATA image comparison after an update.

Syntax:

```
storcli /cx get nvdata file=<filepath>
```

Where:

/c*x* – Specifies the controller, where *x* is the index of the controller, and *filepath* refers to the location of the current binary NVDATA image of the firmware.

Input example:

```
storcli /c1 get nvdata file=c:\fw\nvdata.fw
```

storcli /c*x* get flash file= *<filepath>*

Using this command, you can upload the complete contents of the controller flash to a file specified by you. If the file already exists, this command overwrites the existing file. This command fails if there is not enough disk space to hold the controller flash file.

Syntax:

```
storcli /cx get flash file=<filepath>
```

Where:

/c*x* – Specifies the controller, where *x* is the index of the controller, and *filepath* refers to the location of the controller flash file.

Input example:

```
storcli /c1 get flash file=c:\fw\flash.rom
```

Other Commands

StorCLI also supports the following commands:

```
storcli /cx restart
```

```
storcli /cx/pall show
```

storcli /c*x* restart

Using this command, you can reset a specific controller or reset all controllers connected to the host. This command resets the chip hardware and reinitializes all the chip information. This command also performs the following operations:

- Moves the new firmware image from the backup location to the current location of the firmware.
- Migrates the NVDATA changes.
- Brings up and runs the new firmware.

Syntax:

```
storcli /cx restart
```

Where:

/c*x* – Specifies the controller, where *x* is the index of the controller, and *all* specifies all the controllers connected to the host.

Usage examples:

```
storcli /c0 restart
```

```
storcli /call restart
```

storcli /c*x* /pall show

Using this command, you can list the state of all phy links on a specified adapter.

Syntax:

```
storcli /cx/pall show
```

Where:

`/cx` — Specifies the controller, where x is the index of the controller.

Usage example:

```
storcli /c0/pall show
```

SAS Address Assignment Rule

The PHY SAS address is calculated by incrementing the controller SAS address by one, based on the number of PHYs.

Suppose you are using 16 or 8 PHY cards and four connectors exist: C3, C2, C1, and C0. Each connector has four PHYs, and the autoport configuration is always enabled. Connector C3 has PHYs 0 to 3, Connector C2 has PHYs 4 to 7, Connector C1 has PHYs 8 to 11, and Connector C0 has PHYs 12 to 15.

- If you are connecting four different target devices and want to plug a cable into Connector 1, the SAS address for this port is 0x5000_0000_8000_0008 because the connector's first PHY is 8.
- Furthermore, when you plug a cable into Connector 0, the SAS address for this port is 0x5000_0000_8000_0009.
- Assuming nothing is connected to the HBA and you plug a cable into Connector 0, the SAS address assigned to this port is 0x5000_0000_8000_0008.
- Again, assuming nothing is connected to the HBA and you plug a cable into Connector 3, the SAS address assigned to this port is 0x5000_0000_8000_0000.
- Next, when a cable is plugged into Connector 2, the SAS address assigned to this port is 0x5000_0000_8000_0001.

This logic is also applicable for cards with eight PHYs.

Controllers have two SAS cores; each core can have a wide port, with at the most x8 connections. While connectors C0 and C1 can belong to one core, connectors C2 and C3 can belong to another core.

Revision History

Version 1.5, January 15, 2021

The following document changes were made.

- Added Diagnostic Command.
- Updated Controller Configuration Commands.
- Updated Show and Set Controller Properties Commands.
- Updated Windows Driver RTTrace.
- Updated Enclosure Commands.
- Updated Upgrade, Downgrade, and Recovery Commands.
- Added Enclosure Commands to Supported Commands on Initiator-Target Controllers.
- Updated Drive Commands.

Version 1.4, September 9, 2020

The following document changes were made.

- Updated Other Commands.
- Updated Automated Physical Drive Configurations.
- Updated Flashing Controller Firmware Command while the Firmware Is Operational.
- Added Drive Sanitize Command.
- Updated Automated Physical Drive Configurations.
- Added SPDM Commands.
- Updated Drive Performance Monitoring Commands.
- Updated CacheVault Commands.

Version 1.3, January 23, 2020

The following document changes were made.

- Updated Flashing Controller Firmware Command while the Firmware Is Operational.
- Added SnapDump Commands.
- Updated Drive Secure Erase Commands.
- Updated JBOD Commands.
- Updated Drive Group Show Commands.
- Updated CacheVault Commands.
- Updated Enclosure Commands.
- Updated Automated Physical Drive Configurations.
- Updated Drive Commands.
- Updated Other Commands.

Version 1.2, November 30, 2017

The following document changes were made.

- Updated Supported Operating Systems.
- Updated Show and Set Controller Properties Commands.
- Updated JBOD Commands.

Version 1.1, September 11, 2017

The following document changes were made.

- Updated Upgrade, Downgrade, and Recovery Commands.
- Updated Download Commands.

Preliminary, Version 1.0, October 28, 2016

Initial document release.