2002 Series
CLI Reference Guide

P/N 83-00004522-12
Revision A
August 2009

The material in this document is for information only and is subject to change without notice. While reasonable efforts have been made in the preparation of this document to assure its accuracy, changes in the product design can be made without reservation and without notification to its users.

Contents

Tables

Preface

This guide provides information about managing a R/Evolution 2002 Series storage system by using its command-line interface (CLI).

Intended audience

This guide is intended for storage system administrators.

Prerequisites

Prerequisites for using this product include knowledge of:

- Network administration
- Storage system configuration
- Storage area network (SAN) management and direct attach storage (DAS)
- Fibre Channel, Serial Attached SCSI (SAS), Internet SCSI (iSCSI), and Ethernet protocols

Related documentation

In addition to this guide, please refer to other documents for this product:

- The Installation document for your product model
- R/Evolution 2002 Series Setup Guide
- R/Evolution 2002 Series RAIDar User Guide
- R/Evolution 2002 Series FRU Installation and Replacement Guide
- R/Evolution Event Descriptions
- Online help for R/Evolution 2002 Series management interfaces

See Dot Hill's Customer Resource Center web site for additional information: `http://crc.dothill.com`.

Document conventions and symbols

Table 1 Document conventions

Convention	Element
Navy blue text	Cross-reference links and e-mail addresses
`Navy blue text`	Web site addresses
Bold font	• Key names • Text typed into a GUI element, such as into a box • GUI elements that are clicked or selected, such as menu and list items, buttons, and check boxes
Italics font	Text emphasis
`Monospace font`	• File and directory names • System output • Code • Text typed at the command-line
`Monospace, italic font`	• Code variables • Command-line variables
`Monospace, bold font`	Emphasis of file and directory names, system output, code, and text typed at the command line

△ **CAUTION:** Indicates that failure to follow directions could result in damage to equipment or data.

NOTE: Provides additional information.

1 Using the CLI

This chapter introduces the R/Evolution 2002 Series command-line interface (CLI).

Accessing the CLI

The CLI software embedded in controller modules enables you to manage a storage system out of band. You can access the CLI in two ways:

- By using Telnet, an SSH application, or a terminal emulator on a management host that is remotely connected through a LAN to a controller module's network port. See your product's setup guide for information about setting management port IP addresses using the CLI.
- By using a terminal emulator on a management host that is directly connected to a controller module's serial CLI port.

Table 2 Default usernames and passwords

Username	Password	Access level
`monitor`	`!monitor`	Monitor (view only)
`manage`	`!manage`	Manage (view and change)

CLI output modes

The CLI has two output modes:

- Console mode, which is the human-to-computer interface (HCI).
- API mode, which is the computer-to-computer interface (CCI).

Console mode enables users to interact with the CLI and obtain easily readable information. This mode automatically sizes fields according to content and adjusts content to window resizes. These capabilities would present problems for a CCI in the form of scripts or other client software.

API mode uses XML format and enables any external application to interact with the storage system. The XML format is constructed to permit new fields to be added without impacting existing clients if they follow standard XML parsing conventions. The XML format also contains alternate fields for values which may be localized, such as a state value. These values are guaranteed not to change with different languages and can be used for scripting.

Scripting is not supported using console mode because labels, field sizes, and order of fields may change in future firmware releases. To properly script using the CLI, use API mode. API mode is expected to remain consistent from release to release; field names will be consistent and new functionality will be added as new fields. These types of changes in XML output will not impact a conventional XML parsing engine.

You can change the CLI output mode by using the `set cli-parameters` command; see Example on page 62.

Using CLI interactively

By default the CLI is an interactive application. When you are logged into the CLI, the CLI waits for a command to be entered and then responds to it.

The following example shows interactively starting a Telnet session, logging into the CLI, executing a command to show free (available) disks, and exiting the CLI:

```
$: telnet 172.22.5.55
Login: monitor
Password: ********

product
System Name: Test
System Location: Lab
Version: version

# show disks free
Location  Serial Number        Vendor   Rev  How Used   Type   Size
  Rate(Gb/s)  SP Status
------------------------------------------------------------------------------
0.2       3NM4APFR000098248HBV vendor   rev  AVAIL      SAS    146.8GB
  3.0            OK
------------------------------------------------------------------------------
# exit
```

Using CLI scripts

CLI commands can be scripted using a Telnet client like Expect or a Perl library. The following example shows how to construct a script using a Perl library for Telnet communication.

```
use Net::Telnet;
sub cLogin {
    $telnet->open($_[0]);
    $telnet->waitfor(/(login|username)[: ]*$/i);
    $telnet->print("$_[1]");
    $telnet->waitfor(/password[: ]*$/i);
    $telnet->print("$_[2]");
    # either got a login or a prompt
    @ok = $telnet->waitfor(/(#|login:*) /i);
    if ($debug_commands == 1) { print "-"; print @ok; print "-\n"; }
    if ($ok[1] =~ m/login/gi)
    {
       return 0;
    }
    else
    {
       return 1;
    }
}
$ipAddr = $ARGV[0];
$username = $ARGV[1];
$password = $ARGV[2];
$telnet = new Net::Telnet ( Timeout=>10,
Errmode=>'die',
Prompt => '/\# $/i');
if ( !cLogin($ipAddr, $username, $password) == 1 )
{
    print("Error: $username user failed to log in. Exiting.\n");
    $telnet->close;
    exit(0);
}
```

The above shows a Perl script for logging in. `cLogin` is called at the start of the script to log a user into the CLI. The script uses the command-line parameters specified as the IP address, username, and password. After the user has been logged in, other commands can be sent to the CLI.

For scripting support, change the CLI output mode from the default, console, which produces human-readable output, to API, which produces XML output.

In the following command, the first argument enables API mode to allow easier parsing. The second argument disables the paging mode that pauses for each full screen of command output.

```
$telnet->cmd("set cli-parameters api pager disabled");
```

The following code segment shows how to get the entire configuration information from the CLI and print the output. The output can easily be redirected to a file for archiving.

```
@sV = $telnet->cmd("show configuration");
for ($i=0; $i<scalar(@sV); $i++)
{
print ("@sV[ $i ]");
}
```

The next section provides more information about using the XML API.

Using the XML API

You can use an XML parser, such as `XML::Parser` in Perl, to process the XML output and store this information as objects. The XML parser should use the Document Type Definition (DTD) version that corresponds to the firmware level to ensure that the XML is validated. By obtaining the latest DTD for validation, the parser will be forward compatible.

The output of each CLI command is composed of valid XML data until the CLI prompt (typically #) is encountered. The output contains a valid XML header followed by the XML elements described in the following table.

Table 3 XML API elements

Element	Description and attributes
`RESPONSE`	The `RESPONSE` element is the top-level element, which contains all data output for the CLI command that was issued. The response includes: • A number of `OBJECT` elements, which varies by command. • A status object that provides a message and return code. A return code of 0 indicates that the command succeeded. Any other return code is an error code. There is only one `RESPONSE` element per issued command.
`OBJECT`	In general, an `OBJECT` element describes a storage-system component such as a disk or a volume. An object has these attributes: • `basetype`. This attribute allows output in brief mode to be correlated with metadata to reduce the overhead of each command, as described in XML API optimization. This is also a good field to use to detect the type of the object (e.g., a disk, a volume, etc.). • `name`. The name of the object. • `oid`. The unique identifier for the object in the scope of the response. The `OBJECT` element can contain `PROPERTY` elements.
`PROPERTY`	A `PROPERTY` element provides detail about the attributes of an `OBJECT`. A property has these attributes: • `name`. The unique name for the property within the object. • `type`. The type of data represented by the element data. • `size`. Typically the maximum size of the output. Usually only important if the console output is displayed in rows. • `draw`. Whether to show or hide this data in console format. • `sort`. The type of sorting that can be applied to this property. • `key`. Indicates whether this property is a key value to identify this object. • `display-name`. The label for this data to show in user interfaces.

Table 3 XML API elements (continued)

Element	Description and attributes
COMP	A COMP (composition) element associates nested objects, such as a task object within a schedule object. A composition element has these attributes: • P. The oid of the part component. • G. The oid of the group component. An alternative to using COMP elements is described in XML API optimization.
ASC	The association element provides a simple association description between two objects in the response. • A. First object. • B. Second object.

Scripting guidelines

When writing scripts to parse XML API output, use an XML library to parse the data. For parsing, a script should *not* rely on ordering, spacing, or column position. To find a specific property, a script should compare property names as it searches through the data. This allows the script to be compatible with future versions that could potentially add new fields to the output.

The output of `show` commands is intended for monitoring or obtaining the current configuration. Other commands provide configuration data and display one or more status objects that specify the status of command processing. The last status object specifies the overall status of the command; other status objects indicate intermediate processing status.

The following example shows the XML API status object:

```
<OBJECT basetype="status" name="status" oid="5">
    <PROPERTY name="response-type" type="enumeration" size="12" draw="false"
sort="nosort" display-name="Response Type">Success</PROPERTY>
    <PROPERTY name="response-type-numeric" type="enumeration" size="12"
draw="false" sort="nosort" display-name="Response">0</PROPERTY>
    <PROPERTY name="response" type="string" size="180" draw="true" sort="nosort"
display-name="Response">Command completed successfully.</PROPERTY>
   <PROPERTY name="return-code" type="int32" size="5" draw="false" sort="nosort"
display-name="Return Code">0</PROPERTY>
    <PROPERTY name="component-id" type="string" size="80" draw="false"
sort="nosort" display-name="Component ID"></PROPERTY>
</OBJECT>
```

XML API examples

This section provides example output from the XML API.

The XML API is available through the CLI interface, which can be accessed via Telnet or SSH. The command input is in standard CLI format as defined in this guide. The output is in XML which conforms to the DTD described in XML API DTD content.

The recommended command to enable XML API mode is `set cli-parameters api pager off`, which displays output without pausing after each screenful of data. The following example shows how output of the `show volumes` command appears in XML API mode:

```
# show volumes
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<RESPONSE VERSION="W440">
  <OBJECT basetype="volumes" name="volume" oid="1" format="rows">
    <PROPERTY name="virtual-disk-name" type="string" size="20" draw="true"
sort="string" display-name="Vdisk Name">vd1</PROPERTY>
    <PROPERTY name="volume-name" type="string" size="20" draw="true"
sort="string" display-name="Name">vd1_v0</PROPERTY>
    <PROPERTY name="size" units="GB" type="uint64_t" size="16" draw="true"
sort="size" display-name="Size">35.9GB</PROPERTY>
```

```
    <PROPERTY name="size-numeric" type="uint64_t" size="16" draw="true"
sort="size" display-name="Volumes">70312480</PROPERTY>
    <PROPERTY name="preferred-owner" type="enumeration" size="4" draw="true"
sort="string" display-name="Preferred Owner">A</PROPERTY>
    <PROPERTY name="preferred-owner-numeric" type="enumeration" size="4"
draw="true" sort="string" display-name="Volumes">1</PROPERTY>
    <PROPERTY name="owner" type="enumeration" size="3" draw="true" sort="string"
display-name="Current Owner">A</PROPERTY>
    <PROPERTY name="owner-numeric" type="enumeration" size="3" draw="true"
sort="string" display-name="Volumes">1</PROPERTY>
   <PROPERTY name="serial-number" key="true" type="string" size="33" draw="true"
sort="string" display-name="Serial
Number">00c0ffa000010000f9f99a4801000000</PROPERTY>
    <PROPERTY name="write-policy" type="enumeration" size="13" draw="true"
sort="string" display-name="Cache Write Policy">write-back</PROPERTY>
    <PROPERTY name="write-policy-numeric" type="enumeration" size="13"
draw="true" sort="string" display-name="Volumes">1</PROPERTY>
    <PROPERTY name="cache-optimization" type="enumeration" size="12" draw="true"
sort="string" display-name="Cache Optimization">standard</PROPERTY>
    <PROPERTY name="cache-optimization-numeric" type="enumeration" size="12"
draw="true" sort="string" display-name="Volumes">0</PROPERTY>
    <PROPERTY name="read-ahead-size" type="enumeration" size="16" draw="true"
sort="string" display-name="Read Ahead Size">Default</PROPERTY>
    <PROPERTY name="read-ahead-size-numeric" type="enumeration" size="16"
draw="true" sort="string" display-name="Volumes">-1</PROPERTY>
    <PROPERTY name="volume-type" type="enumeration" size="12" draw="true"
sort="string" display-name="Type">standard</PROPERTY>
   <PROPERTY name="volume-type-numeric" type="enumeration" size="12" draw="true"
sort="string" display-name="Volumes">0</PROPERTY>
    <PROPERTY name="volume-class" type="enumeration" size="8" draw="false"
sort="string" display-name="Class">standard</PROPERTY>
    <PROPERTY name="volume-class-numeric" type="enumeration" size="8"
draw="false" sort="string" display-name="Volumes">0</PROPERTY>
<PROPERTY name="blocks" blocksize="512" type="uint64" size="32" draw="false"
sort="integer" display-name="Blocks">70312480</PROPERTY>
    <PROPERTY name="volume-parent" type="string" size="32" draw="false"
sort="string" display-name="Master Volume"></PROPERTY>
   <PROPERTY name="snap-pool" type="string" size="32" draw="false" sort="string"
display-name="Snap-pool"></PROPERTY>
    <PROPERTY name="virtual-disk-serial" type="string" size="32" draw="false"
sort="string" display-name="Vdisk Serial
Number">00c0ffa000010000bd85994800000000</PROPERTY>
    <PROPERTY name="volume-description" type="string" size="130" draw="true"
sort="string" display-name="Volume Description"></PROPERTY>
  </OBJECT>
<OBJECT basetype="status" name="status" oid="5">
    <PROPERTY name="response-type" type="enumeration" size="12" draw="false"
sort="nosort" display-name="Response Type">Success</PROPERTY>
    <PROPERTY name="response-type-numeric" type="enumeration" size="12"
draw="false" sort="nosort" display-name="Response">0</PROPERTY>
    <PROPERTY name="response" type="string" size="180" draw="true" sort="nosort"
display-name="Response">Command completed successfully.</PROPERTY>
   <PROPERTY name="return-code" type="int32" size="5" draw="false" sort="nosort"
display-name="Return Code">0</PROPERTY>
    <PROPERTY name="component-id" type="string" size="80" draw="false"
sort="nosort" display-name="Component ID"></PROPERTY>
</OBJECT>
</RESPONSE>
```

The following example shows XML API output from the `create vdisk` command:

```
# create vdisk vd-1 disks 2.6,2.7,2.8 level r5
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<RESPONSE VERSION="W440">
<OBJECT basetype="status" name="status" oid="1">
    <PROPERTY name="response-type" type="enumeration" size="12" draw="false"
sort="nosort" display-name="Response Type">Success</PROPERTY>
    <PROPERTY name="response-type-numeric" type="enumeration" size="12"
draw="false" sort="nosort" display-name="Response">0</PROPERTY>
    <PROPERTY name="response" type="string" size="180" draw="true" sort="nosort"
display-name="Response">Command completed successfully. - The vdisk was
created.</PROPERTY>
   <PROPERTY name="return-code" type="int32" size="5" draw="false" sort="nosort"
display-name="Return Code">0</PROPERTY>
    <PROPERTY name="component-id" type="string" size="80" draw="false"
sort="nosort" display-name="Component ID"></PROPERTY>
</OBJECT>
</RESPONSE>
```

XML API DTD content

The following DTD provides the structure of all documents returned by the CLI when XML API mode is enabled. Elements and attributes are described in the table on the following page.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE RESPONSE [
<!-- Copyright Protected Material 2002-2007.  -->
<!--- Response Element. Echoes the request back -->
<!ELEMENT RESPONSE (ASC|COMP|OBJECT|LOG_CONTENT)*>
<!ATTLIST RESPONSE VERSION CDATA #IMPLIED request CDATA #IMPLIED>
<!--- Object Definition. Essentially the object can only contain properties. OID
is unique per object only in each request -->
<!ELEMENT OBJECT (PROPERTY|OBJECT)*>
<!ATTLIST OBJECT oid CDATA #REQUIRED name CDATA #IMPLIED basetype CDATA #IMPLIED
format (pairs|rows|packed|labeled|status) #IMPLIED >
<!--- Property definition -->
<!ELEMENT PROPERTY (#PCDATA)>
<!ATTLIST PROPERTY
    display-name CDATA #IMPLIED
    draw (true|false) #IMPLIED
    size CDATA #IMPLIED
    type (string|uint8|uint16|uint32|uint64|int8|int16|int32|int64|bool|enum)
#IMPLIED
    key (true|false) #IMPLIED
    name CDATA #REQUIRED
    units CDATA #IMPLIED
    blocksize CDATA #IMPLIED
>
<!--- Composition, P is the part component, G is the grouping component -->
<!ELEMENT COMP EMPTY>
<!ATTLIST COMP P IDREF #REQUIRED G IDREF #REQUIRED>
  <!--- Simple Association, A and B are the oids of the Objects -->
<!ELEMENT ASC EMPTY>
<!ATTLIST ASC A IDREF #REQUIRED B IDREF #REQUIRED>
<!ELEMENT LOG_CONTENT (#PCDATA)>
] >
```

XML API basetypes

The following basetypes can be used when calling the `meta` command to obtain complete metadata for all CLI objects.

```
advanced-settings-table
auto-write-through-trigger
cache-parameter
cache-settings
chap-records
cli-parameters
compact-flash
configuration
controller
controllers
controller-cache-parameters
controller-date
cpld-revision
debug-log-parameters
drive-parameters
drives
email-parameters
enclosures
enclosure-list
enclosure-components
enclosure-fru
error
events
expander-ports
fan
fc-port
header
hosts
host-parameters
host-view
host-view-mappings
host-wwn-name
inquiry
io-modules
iscsi-parameters
iscsi-port
job-parameters
license
log-header-table
master-volumes
network-parameters
ntp-status
policy-threshold
port
power-supplies
redundancy
refresh_counters
reset-snapshot-tasks
retained-snapshots
sas-link-health
sas-port
sas-status-controller-a
sas-status-controller-b
schedules
security-communications-protocols
sensors
ses
shutdown-status
snapshots
snapshot-information
snapshot-with-retention-tasks
snap-tasks
snap-pools
snmp-parameters
status
system
system-config
system-parameters-table
tasks
task-details
time-settings-table
unhealthy-component
users
versions
virtual-disks
volume-copy-status
volume-copy-tasks
volume-view
volume-view-mappings
volumes
```

XML API optimization

The following are two ways to optimize XML API performance:

- Use embedded objects. This allows one object to contain not only properties but also other objects. In general, parsing a structure such as this is easier as the association between objects is simpler. This is an alternative to using `COMP` elements.
- Use brief mode. Brief mode, which is enabled by default, shows only the `name` and `key` properties of objects in normal commands. Other properties can be obtained by using the `meta` command with the basetype of the object. This optimization reduces the number of bytes transmitted for each request and allows caching of CLI metadata. Brief mode can be enabled or disabled by using the `set cli-parameters` command.

In the following example, embedded objects contain media-specific detail for ports, and only `name` and `key` properties are shown:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<RESPONSE VERSION="W440">
  <OBJECT basetype="port" name="name" oid="1" format="rows">
    <PROPERTY name="durable-id">hostport_A1</PROPERTY>
    <PROPERTY name="controller" key="true" >A</PROPERTY>
    <PROPERTY name="controller-numeric" key="true" >1</PROPERTY>
    <PROPERTY name="port" key="true" >A1</PROPERTY>
    <PROPERTY name="media">FC(-)</PROPERTY>
    <PROPERTY name="target-id">500C0FF000005000</PROPERTY>
    <PROPERTY name="status">Disconnected</PROPERTY>
    <PROPERTY name="status-numeric">6</PROPERTY>
    <PROPERTY name="actual-speed"></PROPERTY>
```

```
    <PROPERTY name="actual-speed-numeric">255</PROPERTY>
    <PROPERTY name="configured-speed">2Gb</PROPERTY>
    <PROPERTY name="configured-speed-numeric">1</PROPERTY>
    <PROPERTY name="health">N/A</PROPERTY>
    <PROPERTY name="health-numeric">4</PROPERTY>
    <PROPERTY name="health-reason">Host port is down.</PROPERTY>
      <OBJECT basetype="fc-port" name="port" oid="4" format="rows">
              <PROPERTY name="configured-topology">Loop</PROPERTY>
              <PROPERTY name="primary-loop-id"></PROPERTY>
              <PROPERTY name="secondary-loop-id"></PROPERTY>
            </OBJECT>
      </OBJECT>
  <OBJECT basetype="port" name="name" oid="2" format="rows">
    <PROPERTY name="durable-id">hostport_A2</PROPERTY>
    <PROPERTY name="controller" key="true" >A</PROPERTY>
    <PROPERTY name="controller-numeric" key="true" >1</PROPERTY>
    <PROPERTY name="port" key="true" >A2</PROPERTY>
    <PROPERTY name="media">FC(-)</PROPERTY>
    <PROPERTY name="target-id">500C0FF000005100</PROPERTY>
    <PROPERTY name="status">Disconnected</PROPERTY>
    <PROPERTY name="status-numeric">6</PROPERTY>
    <PROPERTY name="actual-speed"></PROPERTY>
    <PROPERTY name="actual-speed-numeric">255</PROPERTY>
    <PROPERTY name="configured-speed">Auto</PROPERTY>
    <PROPERTY name="configured-speed-numeric">3</PROPERTY>
    <PROPERTY name="health">N/A</PROPERTY>
    <PROPERTY name="health-numeric">4</PROPERTY>
    <PROPERTY name="health-reason">Host port is down.</PROPERTY>
      <OBJECT basetype="fc-port" name="port" oid="4" format="rows">
              <PROPERTY name="configured-topology">Loop</PROPERTY>
              <PROPERTY name="primary-loop-id"></PROPERTY>
              <PROPERTY name="secondary-loop-id"></PROPERTY>
      </OBJECT>
  </OBJECT>
  ...
  <OBJECT basetype="status" name="status" oid="9">
      <PROPERTY name="response-type">Success</PROPERTY>
      <PROPERTY name="response-type-numeric">0</PROPERTY>
      <PROPERTY name="response">Command completed successfully.</PROPERTY>
      <PROPERTY name="return-code">0</PROPERTY>
      <PROPERTY name="component-id"></PROPERTY>
  </OBJECT>
</RESPONSE>
```

Command syntax

Keywords and parameters

Command keywords must be entered in lowercase. Parameter values can be entered in uppercase and lowercase.

Unless otherwise specified, a parameter value can include any valid UTF-8 characters except backslash (\), comma, double quote ("), and control characters. A parameter value that includes a space must be enclosed in double quotes.

Parameters such as names of users and volumes have a maximum length in bytes. ASCII characters are 1 byte; most Latin (Western European) characters with diacritics are 2 bytes; most Asian characters are 3 bytes.

Parameters can be entered in any order. However, if the value of a parameter with no keyword is the same as the keyword of an optional parameter, the optional parameter must precede the value. For example, to create a vdisk named `spare`, the `spare` parameter must precede the name value **`spare`**:
`create vdisk level raid5 disks 1.10-12 spare 1.7` **`spare`**

Disks

Disks are specified by enclosure ID and slot number. Enclosure IDs increment from 0. Disk IDs increment from 0 in each enclosure. You can specify:

- A disk. Example: `0.4`
- A hyphenated range of disks. Example: `0.4-7`
- A comma-separated list of individual disks, ranges, or both (with no spaces). Example: `0.4,0.6-9`
- A RAID 10 or 50 vdisk, with disks in sub-vdisks separated by colons (with no spaces). RAID-50 example: `0.1-3:0.4-6:0.7,0.10-11`

Vdisks

You can specify:

- A vdisk by its name or serial number. A unique serial number is automatically assigned when a vdisk is created, and does not change for the life of the vdisk.
- A list of vdisk names or serial numbers separated by commas (with no spaces). Not all commands support lists. List example: `vd1,"My vdisk"`

Volumes

You can specify:

- A volume by its name or serial number. A unique serial number is automatically assigned when a volume is created, and does not change for the life of the volume.
- A list of volume names or serial numbers separated by commas (with no spaces). Not all commands support lists. List example: `vd1_v1,"Vol #1"`

Ports

Controller host ports are specified by controller ID and port number, and are not case sensitive. Controller IDs are A for the upper controller and B for the lower controller. Port IDs increment from 0 in each controller module. You can specify:

- A port ID. Example: `A1`
- A hyphenated range of IDs. Do not mix controller IDs in a range. Example: `b0-b1`
- A comma-separated list of IDs, ranges, or both (with no spaces). Example: `A1,b0-b1`

Command completion, editing, and history

The CLI supports command completion, command editing, and command history.

When entering commands interactively you can abbreviate their names and keywords. For example, you can enter **`sho cl`** to run the `show cli-parameters` command. If you enter too few letters to uniquely identify a keyword, pressing **Tab** will list commands or keywords that match the entered string and redisplays the string so you can complete it. When scripting commands, type commands in full to aid readability.

The history contains the last 30 commands entered in the active CLI session. You can recall a command from the history, edit it, and run it.

Table 4 Keyboard shortcuts for command completion, editing, and history

To	Press
Complete a partially entered keyword	**Tab**
Get previous command from history	**Up Arrow** or **Ctrl+P**
Get next command from history	**Down Arrow** or **Ctrl+N**
Move cursor left	**Left Arrow** or **Ctrl+B**
Move cursor right	**Right Arrow** or **Ctrl+F**
Move back one word	**Esc+B**
Move forward one word	**Esc+F**
Move cursor to start of line	**Ctrl+A**
Move cursor to end of line	**Ctrl+E**
Transpose current and previous character	**Ctrl+T**
Delete current character	**Ctrl+D**
Delete previous character	**Backspace**
Delete word up to cursor	**Ctrl+W**
Delete rest of word	**Esc+D**
Delete text up to cursor	**Ctrl+U**
Delete rest of line	**Ctrl+K**
Convert rest of word to uppercase	**Esc+C**
Convert rest of word to lowercase	**Esc+L**
Enter command and redisplay prompt	**Ctrl+Z**
Refresh input line	**Ctrl+L**

Viewing help

To view brief descriptions of all commands that are available to the user level you logged in as, enter:

```
help
```

To view help for a command and then return to the command prompt, enter:

```
help command-name
```

To view help for a command that you've started to enter, enter `?`; for example:

```
command-name ?
```

To view the information shown in Command syntax above, enter:

```
help syntax
```

To view the information shown in this topic and in Command completion, editing, and history above, enter:

```
help help
```

Size representations

Operating systems usually show volume size in base 2. Disk drives usually show size in base 10. Memory size is always shown in base 2.

In the CLI, the base for entry and display of storage-space sizes can be set per user or per session; see `create user` and `set cli-parameters`. When entering storage-spaces sizes only, either base-2 or base-10 units can be specified.

Table 5 Size representations in base 2 and base 10

Base 2		Base 10	
Unit	**Size in bytes**	**Unit**	**Size in bytes**
KiB (kibibyte)	2^{10} (1,024)	KB (kilobyte)	10^{3} (1,000)
MiB (mebibyte)	2^{20} (1,048,576)	MB (megabyte)	10^{6} (1,000,000)
GiB (gibibyte)	2^{30} (1,073,741,824)	GB (gigabyte)	10^{9} (1,000,000,000)
TiB (tebibyte)	2^{40} (1,099,511,627,776)	TB (terabyte)	10^{12} (1,000,000,000,000)

Event log

A controller enclosure's event log records all events that have occurred in or been detected by the controller modules and encompasses all field-replaceable units (FRUs) in the enclosure.

Each event has one of the following levels, in decreasing severity:

- Critical. Events that might affect data integrity or system stability.
- Warning. Events that do not affect data integrity.
- Informational. Events that show the change of state or configuration changes.

For information about viewing events, see the `show events` command.

2 Categorical list of commands

This chapter helps you find a command within a category of functionally related commands. A command might appear in more than one category.

Table 6 Commands by category

Category	Commands	
CLI and users	create user	set password
	delete user	set prompt
	exit	set user
	help (see Viewing help)	show cli-parameters
	set cli-parameters	show users
Disks and vdisks	abort scrub	set expander-fault-isolation
	abort verify	set led
	clear disk-metadata	set spares
	create vdisk	set vdisk
	delete vdisk	show disks
	dequarantine	show vdisks
	expand vdisk	trust
	rescan	verify vdisk
	scrub vdisk	
Volumes, hosts, and mapping	create host	set volume
	create volume	show cache-parameters
	create volume-set	show host-maps
	delete host	show hosts
	delete volume	show ports
	expand volume	show volume-maps
	map volume	show volumes
	set cache-parameters	unmap volume
	set host-name	
Snapshots	convert master-to-std	delete snapshot-write-data
	convert std-to-master	expand master-volume
	create master-volume	expand snap-pool
	create snap-pool	reset snapshot
	create snapshots	rollback master-volume
	delete all-master-volumes	set snap-pool-policy
	delete all-snapshots	set snap-pool-threshold
	delete master-volume	show master-volumes
	delete snap-pool	show snap-pools
	delete snapshot	show snapshots

Table 6 Commands by category (continued)

Category	Commands	
Volume copy	abort volumecopy	show volumecopy-status
	create volume-copy	volumecopy
Scheduled tasks	create schedule	show schedule-details
	create task	show schedules
	delete schedule	show task-details
	delete task	show tasks
Event notification	set email-parameters	show events
	set snmp-parameters	show snmp-parameters
	show email-parameters	test
System configuration and utilities	clear cache	show controller-date
	create chap-record	show controllers
	delete chap-records	show disk-parameters
	ping	show enclosures
	reset host-link	show enclosure-status
	restart	show expander-status
	set auto-write-through-trigger (alias: set awt)	show frus
	set chap-record	show host-parameters
	set controller-date	show iscsi-parameters
	set disk-parameters	show job-parameters
	set enclosure	show license
	set expander-fault-isolation	show network-parameters
	set expander-phy	show ntp-status
	set host-parameters	show ports
	set iscsi-parameters	show protocols
	set job-parameters	show redundancy-mode
	set led	show sas-link-health
	set network-parameters	show sensor-status
	set protocols	show shutdown-status
	set system	show system
	show auto-write-through-trigger (alias: show awt)	show system-parameters
	show chap-records	shutdown
	show configuration	stty
		versions (alias: show versions)

Table 6 Commands by category (continued)

Category	Commands	
Service utilities	`clear events`	`set expander-fault-isolation`
	`clear expander-status`	`set expander-phy`
	`restore defaults`	`show debug-log-parameters`
	`set debug-log-parameters`	`show expander-status`
API specific	`meta`	`show advanced-settings`
	`set advanced-settings`	`show refresh-counters`

3 Alphabetical list of commands

This chapter is organized to help you find a command by name. Each command topic includes one or more of the following sections:

Description The command's purpose and notes about its usage

Syntax The command's syntax

Parameters Descriptions of the command's parameters

Output For `show` commands only, descriptions of output fields

Examples One or more examples of the command's usage, if the command has parameters or detailed output

See also Cross-references to commands that are used with the command

abort scrub

Description Aborts the `scrub vdisk` operation for specified vdisks.

Syntax `abort scrub vdisk` *`vdisks`*

Parameters *`vdisks`*
Names or serial numbers of the vdisks to stop scrubbing.

Example Abort scrubbing vdisk `vd1`:

```
# abort scrub vdisk vd1
Info: Scrub aborted on vdisk vd1
Success: Command completed successfully
```

See also
- `scrub vdisk`
- `show vdisks` (to find disks being scrubbed)

abort verify

Description Aborts the `verify vdisk` operation for specified vdisks.

Syntax `abort verify vdisk` *`vdisks`*

Parameters *`vdisks`*
Names or serial numbers of the vdisks to stop verifying.

Example Abort verifying vdisk `vd1`:

```
# abort verify vdisk vd1
Info: Verify aborted on vdisk vd1
Success: Command completed successfully
```

See also
- `show vdisks` (to find disks being verified)
- `verify vdisk`

abort volumecopy

Description Aborts copying a volume. When the abort is complete, the destination volume is deleted.

Syntax `abort volumecopy volume`

Parameters *volume*
Name or serial number of the source or destination volume.

Example Abort creating destination volume `vd1_copy`:

```
# abort volumecopy v1_copy
Success: Command completed successfully. - The volume copy was aborted.
```

See also
- show volumecopy-status
- show volumes
- volumecopy

clear cache

Description Clears unwritable cache data from both controllers. This data cannot be written to disk because it is associated with a volume that no longer exists or whose disks are not online. If the data is needed, the volume's disks must be brought online. If the data is not needed it can be cleared, in which case it will be lost and data will differ between the host and disk. Unwritable cache is also called orphan data.

You can clear unwritable cache data for a specified volume or for all volumes.

Syntax `clear cache [volume volume]`

Parameters volume *volume*
Optional. Name or serial number of the volume whose cache data should be cleared. For syntax, see Command syntax. If this parameter is omitted, the command clears any unneeded orphaned data for volumes that are no longer online or that no longer exist.

Example Clear the cache in both controllers for volume `V1`:

```
# clear cache volume v1
Success: Command completed successfully
```

clear disk-metadata

Description Clears metadata from “leftover” disks. Each disk contains metadata that the system uses to identify the disk's owning vdisk, if any. If the system cannot locate the vdisk, as when the disk has been moved to a different system, the owning vdisk is shown as Leftover. A leftover disk’s Power/Activity/Fault LED is illuminated amber. You must clear the metadata before you can use the disk in a different vdisk or as a spare.

If you specify a disk that is not a leftover, the command will not clear that disk's metadata.

Syntax `clear disk-metadata disks`

Parameters *disks*
IDs of the disks to clear metadata from. For syntax, see Command syntax.

Example Clear metadata for the first enclosure's first disk, which is *not* part of a vdisk:

```
# clear disk-metadata 0.0
Updating disk list...
Info: Command completed successfully. (0.0) - Disk 0.0 metadata was cleared.
Success: Command completed successfully. (0.0) - Metadata was cleared.
```

Clear metadata for disk 0.3, which is part of a vdisk:

```
# clear disk-metadata 0.3
Error: The specified disk is not a leftover disk. (0.3) - Disk 0.3 metadata was
NOT cleared.
Error: The specified disk is not a leftover disk. (0.3) - Metadata was not
cleared for one or more disks.
```

clear events

Description **For use by or with direction from a service technician.**

Clears the event log for controller A, B, or both.

Syntax `clear events [a|b|both]`

Parameters `a|b|both`
Optional. The controller event log to clear. If this parameter is omitted, both event logs are cleared.

Example Clear the event log for controller A:

```
# clear events a
Success: Command completed successfully. - Controller A event log was
successfully cleared.
```

See also
- `show events`

clear expander-status

Description **For use by or with direction from a service technician.**

Clears the counters and status for SAS expander lanes. Counters and status can be reset to a good state for all enclosures, or for a specific enclosure whose status is `Error` as shown by the `show expander-status` command.

NOTE: If a rescan is in progress, the clear operation will fail with an error message saying that an EMP does exist. Wait for the rescan to complete and then retry the clear operation.

Syntax `clear expander-status [enclosure` *`ID`*`]`

Parameters `enclosure` *`ID`*
Optional. The enclosure number.

Example Clear the expander status for the first enclosure:

```
# clear expander-status enclosure 0
Success: Command completed successfully. - Expander status was cleared.
```

See also
- `show expander-status`

convert master-to-std

Description Converts a specified master volume into a standard volume; that is, it disables the volume from accepting snapshots. If the specified volume has associated snapshots, you must delete the snapshots before converting the volume.

Syntax
```
convert master-to-std master-volume
```

Parameters

master-volume
Name or serial number of the master volume to convert. For syntax, see Command syntax.

Example Convert a master volume having no snapshots to a standard volume:

```
# convert master-to-std MV1
Success: Command completed successfully. - The conversion of a master volume to a
standard volume completed.
```

See also
- delete all-snapshots
- show master-volumes

convert std-to-master

Description Converts a standard volume to a master volume; that is, it enables the volume for snapshots and associates it with an existing snap pool. The standard volume and the snap pool must be owned by the same controller, though they can be in different vdisks.

Syntax
```
convert std-to-master snap-pool snap-pool standard-volume
```

Parameters

snap-pool *snap-pool*
Name or serial number of the snap pool to associate with the new master volume. For syntax, see Command syntax.

standard-volume
Name or serial number of the standard volume to convert. For syntax, see Command syntax.

Example Convert standard volume V1 to a master volume and associate it with snap pool SP1:

```
# convert std-to-master snap-pool SP1 V1
Success: Command completed successfully. - The conversion of a standard volume to
a master volume completed.
```

See also
- show volumes

create chap-record

Description For iSCSI, creates a CHAP record to authenticate login requests. When CHAP is enabled, the record enables authentication between the originator (initiator) and recipient (target) of a login request. This command is permitted whether or not CHAP is enabled.

The CHAP record can specify one name-secret pair to authenticate the originator only (one-way CHAP) or two pairs to authenticate both the originator and the recipient (mutual CHAP).

For a login request from an iSCSI host to a storage system, the host is the originator and the storage system is the recipient.

Syntax
```
create chap-record
   name originator-name
   secret originator-secret
   [mutual-name recipient-name mutual-secret recipient-secret]
```

Parameters

name *originator-name*
The originator name; typically the originator's IQN. The name can have a maximum of 223 bytes, including 0–9, lowercase a–z, hyphen, colon, and period.

`secret` *originator-secret*
The secret that the recipient uses to authenticate the originator. The secret is case sensitive and can include 12–16 bytes.

`mutual-name` *recipient-name*
Optional; for mutual CHAP only. The recipient name; typically the recipient's IQN. The name can have a maximum of 223 bytes, including 0–9, lowercase a–z, hyphen, colon, and period. A storage system's IQN is a controller host-port name without its `.a` or `.b` suffix; see the Port-Name field shown by the `show host-parameters` command. This parameter and `mutual-secret` must be set together.

`mutual-secret` *recipient-secret*
Optional; for mutual CHAP only. The secret that the originator uses to authenticate the recipient. The secret is case sensitive, can include 12–16 bytes, and must differ from the originator secret. A storage system's secret is shared by both controllers. This parameter and `mutual-name` must be set together.

Example Create a one-way CHAP record to enable a storage system to authenticate a host initiator:

```
# create chap-record name iqn.1991-05.com.microsoft:myHost.domain secret
123456abcDEF
Success: created CHAP record for iqn.1991-05.com.microsoft:myHost.domain
```

See also
- `delete chap-records`
- `set chap-record`
- `show chap-records`
- `show host-parameters`
- `show iscsi-parameters`

create host

Description Creates a host entry with an associated nickname. When mapping volumes to hosts the nickname can make a host easy to recognize.

Syntax `create host id` *ID nickname*

Parameters `id` *ID*
For FC and SAS, the host node's 16-hex-digit WWPN. For iSCSI, the initiator's IQN. A host ID cannot have more than one entry in the system.

nickname
A nickname for the host node. The name is case sensitive, cannot include a comma, double quote, or backslash, and can have a maximum of 15 bytes. A name that includes a space must be enclosed in double quotes.

Example Create an entry named `Host1` for an FC host whose WWPN is 207000C0FF001122:

```
# create host id 207000C0FF001122 Host1
Success: Command completed successfully. - The new host was created.
```

See also
- `show hosts`

create master-volume

Description Creates a volume that is enabled for snapshots. The master volume is created in a specified vdisk and is associated with a specified snap pool. The vdisk and snap pool must be owned by the same controller.

Syntax

```
create master-volume
    vdisk vdisk
    size size[B|KB|MB|GB|TB|KiB|MiB|GiB|TiB]
    snap-pool snap-pool
    [lun LUN]
    name
```

Parameters

vdisk *vdisk*
Name or serial number of the vdisk to create the volume in. For syntax, see Command syntax.

size *size*[B|KB|MB|GB|TB|KiB|MiB|GiB|TiB]
Sets the volume size using the current base, as shown by `show cli-parameters`. The unit can be specified as follows:

- If base 2 is in use: `B` (bytes), `KiB` (kibibytes), `MiB` (mebibytes), or `GiB` (gibibytes)
- If base 10 is in use: `B` (bytes), `KB` (kilobytes), `MB` (megabytes), or `GB` (gigabytes)

If no unit is specified, the unit is 512-byte blocks.

snap-pool *snap-pool*
Name or serial number of the snap pool to associate with the new master volume. For syntax, see Command syntax.

lun *LUN*
Optional. A default LUN to assign to the new master volume. If this parameter is omitted, no LUN is assigned.

name
A name for the new volume. The name is case sensitive, cannot include a comma, double quote, or backslash, and can have a maximum of 20 bytes. A name that includes a space must be enclosed in double quotes.

Example Create the 20-GB master volume `MV1` on vdisk `VD1`, and associate it with snap pool `SP1`:

```
# create master-volume vdisk VD1 size 20GB snap-pool SP1 lun 3 MV1
Success: Command completed successfully. - The master volume was created.
```

See also

- `show master-volumes`
- `show snap-pools`
- `show vdisks`

create schedule

Description Schedules a task to run automatically.

Syntax
```
create schedule
    schedule-specification "specification"
    task-name task-name
    schedule-name
```

Parameters

`schedule-specification "specification"`
Defines when the task will first run, and optionally when it will recur and expire. You can use a comma to separate optional conditions. Dates cannot be in the past.

- `start mm/dd/yyyy hh:mm [AM|PM]`
  If neither `AM` nor `PM` is specified, a 24-hour clock is used. If you use the `between` condition, below, the start time must be in the between range.

Optional conditions:

- `every # minutes|hours|days|weeks|months|years`
- `between hh:mm [AM|PM] and hh:mm [AM|PM]`
- `only any|first|second|third|fourth|fifth|last|#st|#nd|#rd|#th weekday|weekendday|Sunday|Monday|Tuesday|Wednesday|Thursday|Friday|Saturday of year|month|January|February|March|April|May|June|July|August|September|October |November|December`
- `count #`
- `expires mm/dd/yyyy hh:mm [AM|PM]`

`task-name task-name`
The task to run. The name is case sensitive.

schedule-name
A name for the new schedule. The name is case sensitive, cannot include a comma, double quote, or backslash, and can have a maximum of 32 bytes. A name that includes a space must be enclosed in double quotes.

Example Create schedule `Sched1` that runs `Task1` for the first time on March 1, 2007; runs daily between midnight and 1:00 AM; and runs for the last time in the morning of January 1, 2008:

```
# create schedule schedule-specification "start 3/1/2007 00:01, every 1 days,
between 12:00 AM and 1:00 AM, expires 1/1/2008 1:00 AM" task-name Task1 Sched1
Success: Command completed successfully. - The schedule was created.
```

Create schedule `Sched2` that runs `Task2` for the first time on March 1, 2007, and on the first weekday of each month, with no expiration:

```
# create schedule schedule-specification "start 3/1/2007 00:01 only first weekday
of month" task-name Task2 Sched2
Success: Command completed successfully. - The schedule was created.
```

See also

- show schedules
- show task-details
- show tasks

create snap-pool

Description Creates a snap pool to use for snapshot data. A snap pool is an internal volume and cannot be mapped.

Syntax
```
create snap-pool
    vdisk vdisk
    size size[B|KB|MB|GB|TB|KiB|MiB|GiB|TiB]
    name
```

Parameters

`vdisk vdisk`
Name or serial number of the vdisk to create the snap pool in. For syntax, see Command syntax.

`size size[B|KB|MB|GB|TB|KiB|MiB|GiB|TiB]`
Sets the volume size using the current base, as shown by `show cli-parameters`. The unit can be specified as follows:

- If base 2 is in use: `B` (bytes), `KiB` (kibibytes), `MiB` (mebibytes), or `GiB` (gibibytes)
- If base 10 is in use: `B` (bytes), `KB` (kilobytes), `MB` (megabytes), or `GB` (gigabytes)

If no unit is specified, the unit is 512-byte blocks. The minimum size is 10 GB/GiB.

name
A name for the new snap pool. The name is case sensitive, cannot include a comma, double quote, or backslash, and can have a maximum of 20 bytes. A name that includes a space must be enclosed in double quotes.

Example Create the 20-GB snap pool `SP1` on vdisk `VD1`:

```
# create snap-pool vdisk VD1 size 20GB SP1
Success: Command completed successfully. - The snap-pool was created.
```

See also
- `show snap-pools`
- `show vdisks`

create snapshots

Description Creates a snapshot of each specified source volume. The source volume can be a standard volume or a master volume. The first time a snapshot is created of a standard volume, the volume is converted to a master volume and a snap pool is created. The snap pool's size is 20% of the volume size or 10 GB, whichever is larger. Before creating or scheduling snapshots, verify that the vdisk has enough free space to contain the snap pool.

Syntax
```
create snapshots
    volumes volumes
    [master-volumes master-volumes]
    snap-names
```

Parameters

volumes *volumes*
A comma-separated list of standard or master volumes to take snapshots of. A standard volume is converted to a master volume before a snapshot is taken. For syntax, see Command syntax.

`master-volumes master-volumes`
Deprecated; use the volumes parameter.

snap-names
A comma-separated list of names for the resulting snapshots. A name is case sensitive, cannot include a comma, double quote, or backslash, and can have a maximum of 20 bytes. A name that includes a space must be enclosed in double quotes.

Example Create snapshots of standard volume `V1` and master volume `V2`:

```
# create snapshots volumes V1,V2 V1snap,V2snap
Success: Command completed successfully. (V1snap) - Snapshot(s) were created.
```

See also
- show snapshots
- show volumes

create task

Description Creates a task that can be scheduled. You can create a task to take a snapshot of a master volume, to copy a snapshot or a master volume to a new standard volume, or to reset a snapshot.

△ **CAUTION:** Before scheduling a reset snapshot task, consider that if the snapshot is mounted to a host operating system, the snapshot must be unmounted before the reset is performed; leaving it mounted can cause data corruption. You should create a scheduled job on the host to unmount the snapshot prior to resetting the snapshot.

Syntax To create a task to take a snapshot:

```
create task type TakeSnapshot master-volume volume snapshot-prefix prefix
retention-count # name
```

To create a task to reset a snapshot:

```
create task type ResetSnapshot snapshot-volume volume name
```

To create a task to copy a volume:

```
create task type VolumeCopy source-volume volume dest-vdisk vdisk dest-prefix
prefix [modified-snapshot yes|no] name
```

Parameters `type TakeSnapshot|ResetSnapshot|VolumeCopy`
The task type:

- `TakeSnapshot`: Takes a snapshot of a master volume.
- `ResetSnapshot`: Deletes the data in the snapshot and resets it to the current data in the associated master volume. The snapshot's name and other volume characteristics are not changed.

  △ **CAUTION:** Before scheduling a reset snapshot task, consider that if the snapshot is mounted to a host operating system, the snapshot must be unmounted before the reset is performed; leaving it mounted can cause data corruption.

- `VolumeCopy`: Copies a snapshot or a master volume to a new standard volume. The command creates the destination volume you specify, which must be in a vdisk owned by the same controller as the source volume.

`master-volume` *volume*
Name or serial number of the volume to take a snapshot of. For syntax, see Command syntax.

`snapshot-prefix` *prefix*
Optional. A label to identify snapshots created by this task. Snapshot names have the format *prefix*`_s001` through *prefix*`_s1023`.

`retention-count #`
Optional. The number of snapshots with this prefix to retain. When a new snapshot exceeds this limit, the oldest snapshot with the same prefix is deleted.

`snapshot-volume` *volume*
Name or serial number of the snapshot to reset. For syntax, see Command syntax.

`source-volume` *volume*
Name or serial number of the master volume or snapshot to copy. For syntax, see Command syntax.

`dest-vdisk` *vdisk*
Name or serial number of the destination vdisk for the volume copy. For syntax, see Command syntax.

`dest-prefix` *prefix*
Optional. A label to identify the volume copy created by this task. Copy names have the format *prefix*`_c001` through *prefix*`_c1023`.

`modified-snapshot yes|no`
Optional. Specifies whether to include or exclude modified write data from the snapshot in the copy. This parameter applies only when the source volume is a snapshot; it is ignored if the source volume is a master volume.

- `yes`: Include modified snapshot data.
- `no`: Exclude modified snapshot data.

If this parameter is omitted for a snapshot, modified snapshot data is excluded.

name
A name for the new task. The name is case sensitive, cannot include a comma, double quote, or backslash, and can have a maximum of 32 bytes. A name that includes a space must be enclosed in double quotes.

Example Create task `Task1` that takes a snapshot of master volume `VD1_V1` and retains only the latest four snapshots with the prefix `VD1_V1` (e.g., `VD1_V1_S0001`):

```
# create task type TakeSnapshot master-volume VD1_V1 snapshot-prefix VD1_V1
retention-count 4 Task1
Success: Command completed successfully. - The task was created.
```

Create task `Task2` that resets snapshot `VD1_S0001`:

```
# create task type ResetSnapshot snapshot-volume VD1_S0001 Task2
Success: Command completed successfully. - The task was created.
```

Create task `Task3` that copies volume `VD1_V1` to vdisk `VD2` with name `C_V0001`:

```
# create task type VolumeCopy source-volume VD1_V1 dest-vdisk VD2 dest-prefix C
modified-snapshot yes Task3
Success: Command completed successfully. - The task was created.
```

See also
- `create schedule`
- `show task-details`
- `show tasks`
- `show volumes`

create user

Description Creates a user profile. The system supports 12 user profiles.

Syntax

```
create user
    [base 2|10]
    [interfaces values]
    [level monitor|manage]
    [locale English|en]
    [password password]
    [precision #]
    [storage-size-base 2|10]
    [storage-size-precision #]
    [storage-size-units auto|MB|GB|TB]
    [temperature-scale celsius|c|fahrenheit|f]
    [timeout #]
    [type standard|advanced|diagnostic]
    [units auto|MB|GB|TB]
    name
```

Parameters `base 2|10`

Optional. Sets the base for entry and display of storage-space sizes:

- `2`: Sizes are shown as powers of 2, using 1024 as a divisor for each magnitude.
- `10`: Sizes are shown as powers of 10, using 1000 as a divisor for each magnitude. This is the default.

Operating systems usually show volume size in base 2. Disk drives usually show size in base 10. Memory size is always shown in base 2.

`interfaces` *`values`*

Optional. Specifies the interfaces that the user can access. Multiple values must be separated by commas and no spaces. The defaults are `cli` and `wbi`.

- `cli`: Command-line interface.
- `wbi`: Web-browser interface.
- `ftp`: File transfer protocol interface.
- `none`: No interfaces.

`level monitor|manage`

Optional.

- `monitor`: User can view but not change system settings. This is the default.
- `manage`: User can view and change system settings.

`locale English|en`

Optional. The display language. The default is English.

`password` *`password`*

Optional. A password is case sensitive and can include a maximum of 19 bytes, using characters except a backslash, double quote, or space. If this parameter is omitted, the command prompts you to enter and re-enter a password for the user.

`precision #`

Optional. Sets the number of decimal places (1–10) for display of storage-space sizes. Default is 1.

`storage-size-base 2|10`

Optional. Alias for `base`.

`storage-size-precision #`

Optional. Alias for `precision`.

`storage-size-units auto|MB|GB|TB`

Optional. Alias for `units`.

`temperature-scale celsius|c|fahrenheit|f`

Optional. Specifies to use the Celsius scale or Fahrenheit scale for temperature values. Default is Celsius.

`timeout #`

Optional. Sets the timeout value in seconds for the login session. Valid values are 30–9999, where 9999 means do not timeout. The default is 1800 seconds (30 minutes).

`type standard|advanced|diagnostic`

Optional. Specifies the user's level of technical expertise, to control access to functions in RAIDar.

- `standard`: Enables access to standard administrative functions. This is the default for `monitor` users.
- `advanced`: Enables access to standard and advanced functions. This is the default for `manage` users.
- `diagnostic`: Enables access to standard, advanced, and troubleshooting functions.

`units auto|MB|GB|TB`
Optional. Sets the unit for display of storage-space sizes. `auto` lets the system determine the proper unit for a size. Based on the `precision` setting, if the selected unit is too large to meaningfully display a size, the system uses a smaller unit for that size. For example, if the unit is set to `TB` and the precision is set to 1, the size 0.11709 TB is shown as 119.9 GB. Default is `auto`.

name
A name for the new user, which cannot already exist in the system. The name is case sensitive, cannot include a comma, double quote, or backslash, and can have a maximum of 19 bytes. A name that includes a space must be enclosed in double quotes.

Example Create user `John` who will view system information using base 2 in RAIDar:

```
# create user base 2 interfaces wbi level monitor John
Enter Password for new user John:****
Re-enter Password:****
Info: level: monitor.
Info: interfaces: WBI
Info: The 'type' option was not specified; defaulting to 'standard'.
Info: The 'locale' option was not specified; defaulting to 'English'.
Info: base: 2.
Info: The 'precision' option was not specified; defaulting to '1'.
Info: The 'units' option was not specified; defaulting to 'auto'.
Info: The 'temperature-scale' option was not specified; defaulting to 'celsius'.
Info: The 'timeout' option was not specified; defaulting to '1800' seconds (30
minutes).
Success: Command completed successfully. - The new user was created.
```

See also
- `set user`
- `show users`

create vdisk

Description Creates a vdisk using the specified RAID level, disks, and spares. All disks used in a vdisk and its spares must be either SAS or SATA; mixing disk types is not supported.

For each RAID level, the minimum and maximum numbers of disks supported are:

- NRAID: 1
- RAID 0: 2–16
- RAID 1: 2
- RAID 3: 3–16
- RAID 5: 3–16
- RAID 6: 4–16
- RAID 10: 4–16
- RAID 50: 6–32

Syntax

```
create vdisk
    level nraid|raid0|r0|raid1|r1|raid3|r3|raid5|r5|raid6|r6|raid10|r10
    |raid50|r50
    disks disks
    [assigned-to a|b|auto]
    [spare disks]
    [chunk-size 16k|32k|64k]
    [mode online|offline]
    name
```

Parameters `level nraid|raid0|r0|raid1|r1|raid3|r3|raid5|r5|raid6|r6|raid10|r10|raid50|r50`
Specifies the RAID level.

`disks` *disks*
IDs of the disks to include in the vdisk. RAID 10 requires a minimum of two RAID-1 sub-vdisks each having two disks. RAID 50 requires a minimum of two RAID-5 sub-vdisks each having three disks. For syntax, see Command syntax.

`assigned-to a|b|auto`
Optional. The controller to own the vdisk. To have the system automatically load-balance vdisks between controllers, use `auto` or omit this parameter.

`spare` *`disks`*
Optional. IDs of 1–4 dedicated spares to assign to a RAID 1, 3, 5, 6, 10, or 50 vdisk. For syntax, see Command syntax.

`chunk-size 16k|32k|64k`
Optional. The amount of contiguous data, in KB, that is written to a vdisk member before moving to the next member of the vdisk. For RAID 50, this option sets the chunk size of each RAID-5 subvdisk. The chunk size of the RAID-50 vdisk is calculated as: *configured-chunk-size* x (*subvdisk-members* - 1). The default is 64k.

`mode online|offline`
Optional. Specifies whether the vdisk is initialized online or offline.

- `online`: Enables you to use the vdisk immediately after creating it while it is initializing. Because online uses the verify method to create the vdisk, it takes longer to complete initializing than offline. Online initialization is fault tolerant. This option is the default.
- `offline`: You must wait for the vdisk initialization process to finish before using the vdisk; however, offline takes less time to complete initializing than online. At the time of creation, a vdisk using offline initialization can have either one volume or none. If you want the vdisk to have more than one volume, create the vdisk with no volumes and then add volumes after initialization is complete.

`name`
A name for the new vdisk. The name is case sensitive, cannot include a comma, double quote, or backslash, and can have a maximum of 17 bytes. A name that includes a space must be enclosed in double quotes.

Example Create the RAID-1 vdisk `VD1` using two disks in the first enclosure:

```
# create vdisk level raid1 disks 0.1,0.3 VD1
Success: Vdisk created.
```

Create the RAID-50 vdisk `VD2` having three RAID-5 sub-vdisks, each having three disks:

```
# create vdisk level r50 disks 0.1-3:0.4-6:0.7-9 VD2
Success: Vdisk created.
```

See also
- `set vdisk`
- `show disks`
- `show vdisks`

`create volume`

Description Creates a volume in a vdisk. You can specify a size and name for the volume, and map it to hosts.

Syntax

```
create volume
    vdisk vdisk
    size size[B|KB|MB|GB|TB|KiB|MiB|GiB|TiB]
    [mapping ports.LUN]
    [access read-write|rw|read-only|ro|no-access]
    [lun LUN]
    [ports ports]
    name
```

Parameters `vdisk` *`vdisk`*
Name or serial number of the vdisk to create the volume in. For syntax, see Command syntax.

`size` *size*`[B|KB|MB|GB|TB|KiB|MiB|GiB|TiB]`
Sets the volume size using the current base, as shown by `show cli-parameters`. The unit can be specified as follows:

- If base 2 is in use: `B` (bytes), `KiB` (kibibytes), `MiB` (mebibytes), or `GiB` (gibibytes)
- If base 10 is in use: `B` (bytes), `KB` (kilobytes), `MB` (megabytes), or `GB` (gigabytes)

If no unit is specified, the unit is 512-byte blocks.

`mapping` *ports.LUN*
Optional. The ports and LUN to use for all hosts that are not explicitly mapped (called the default mapping). For syntax, see Command syntax. If this argument is omitted, the volume is unmapped and its LUN is set to `None`. (You can add or remove mappings by using `map volume` and `unmap volume`.)

`access read-write|rw|read-only|ro|no-access`
Optional. The access permission for hosts connected to the controller for this volume: read-write (`rw`), read-only (`ro`), or no-access. When a volume is created with no access, the volume is masked. The default is `read-write`.

`lun` *LUN*
Optional if the access parameter is set to `no-access`. Specifies the LUN to assign to the mapping on all ports. If this parameter is omitted, the default LUN is presented. You cannot use both this parameter and the `mapping` parameter.

`ports` *ports*
Optional. The ports through which the host can access the volume. For syntax, see Command syntax.

name
A name for the new volume. The name is case sensitive, cannot include a comma, double quote, or backslash, and can have a maximum of 20 bytes. A name that includes a space must be enclosed in double quotes.

Example Create the 20-GB volume `V1` on vdisk `VD1`, and map it to ports A1 and B1 using LUN 5:

```
# create volume vdisk VD1 size 20GB ports a1,b1 lun 5 V1
Info: Command completed successfully. (V1) - The volume was created.
```

See also

- `set volume`
- `show vdisks`
- `show volumes`

create volume-copy

See `volumecopy`.

create volume-set

Description Creates multiple volumes in the specified vdisk. The volumes have the same base name, size, and default mapping settings (LUN, access, and ports)

Syntax

```
create volume-set
    vdisk vdisk
    basename base-name
    count #
    size size[B|KB|MB|GB|TB|KiB|MiB|GiB|TiB]
    [baselun base-LUN]
    [access read-write|rw|read-only|ro|no-access]
    [ports ports]
```

Parameters `vdisk` *`vdisk`*
Name or serial number of the vdisk to create the volumes in. For syntax, see Command syntax.

`basename` *`base-name`*
A base name for the new volumes. A name is case sensitive, cannot include a comma, double quote, or backslash, and can have a maximum of 20 bytes. A name that includes a space must be enclosed in double quotes.

Resulting volumes are numbered sequentially from 000 (not 0000) through 1023. If volumes with the specified basename already exist, names of new volumes start with the first available name in the sequence. For example: for basename `vd1_v`, if `vd1_v000` and `vd1_v002` exist, the next volumes created will be `vd1_v001` and `vd1_v003`.

`count #`
The number of volumes to create.

`size` *`size`*`[B|KB|MB|GB|TB|KiB|MiB|GiB|TiB]`
Sets the volume size using the current base, as shown by `show cli-parameters`. The unit can be specified as follows:

- If base 2 is in use: `B` (bytes), `KiB` (kibibytes), `MiB` (mebibytes), or `GiB` (gibibytes)
- If base 10 is in use: `B` (bytes), `KB` (kilobytes), `MB` (megabytes), or `GB` (gigabytes)

If no unit is specified, the unit is 512-byte blocks.

`baselun` *`base-LUN`*
Optional. A LUN to assign to the mapping on all ports. If this parameter is omitted, the default LUN is presented.

`access read-write|rw|read-only|ro|no-access`
Optional. Access privilege that hosts have to these volumes: read-write (`rw`), read-only (`ro`), or no-access. A volume mapped with `no-access` is masked. The default is `read-write`.

`ports` *`ports`*
Optional. The controller ports through which hosts can access the volumes. If not all ports are specified, the unspecified ports are automatically mapped to no access. For syntax, see Command syntax.

Example Create three 20-GB volumes with the base name `vd1_v` in vdisk `vd1`:

```
# create volume-set count 3 size 20GB vdisk vd1 basename vd1_v
Info: Command completed successfully. (vd1_v000) - Created volume vd1_v000.
Info: Command completed successfully. (vd1_v001) - Created volume vd1_v001.
Info: Command completed successfully. (vd1_v002) - Created volume vd1_v002.
```

See also
- `map volume`
- `set volume`
- `show vdisks`
- `show volumes`
- `unmap volume`

delete all-master-volumes

Description Deletes all master volumes associated with a snap pool.

NOTE: You must delete all snapshots that exist for the master volumes before you can delete the master volumes.

Syntax `delete all-master-volumes snap-pool` *`volume`*

Parameters `snap-pool` *`volume`*
Name or serial number of the snap pool whose master volumes should be deleted. For syntax, see Command syntax.

Example Delete all master volumes associated with snap pool `SP1`:

```
# delete all-master-volumes snap-pool SP1
Success: All Master Volumes Deleted.
```

See also
- `delete all-snapshots`
- `show master-volumes`
- `show snap-pools`

delete all-snapshots

Description Deletes all snapshots of a specified volume. All data associated with the snapshots is deleted and associated space in the snap pool is freed for use.

Syntax `delete all-snapshots volume` *`volume`*

Parameters `volume` *`volume`*
Name or serial number of the volume to delete snapshots of. For syntax, see Command syntax.

Example Delete all snapshots associated with master volume `MV1`:

```
# delete all-snapshots volume MV1
Success: All Snapshots Deleted.
```

See also
- `show snapshots`
- `show volumes`

delete chap-records

Description For iSCSI, deletes a specified CHAP record or all CHAP records. This command is permitted whether or not CHAP is enabled.

Syntax To delete the CHAP record for a specific originator:

```
delete chap-records name originator-name
```

To delete all CHAP records:

```
delete chap-records all
```

Parameters `name` *`originator-name`*
The originator name; typically the originator's IQN.

`all`
Delete all CHAP records in the database.

Example Delete the CHAP record for a specific originator:

```
# delete chap-records name iqn.1991-05.com.microsoft:myHost.domain
Success: Command completed successfully.
(iqn.1991-05.com.microsoft:myHost.domain) - A CHAP record was deleted for
iqn.1991-05.com.microsoft:myHost.domain.
```

See also

- create chap-record
- set chap-record
- show chap-records
- show host-parameters

delete host

Description Deletes a manually created host. Does not delete hosts that were discovered or are mapped. Before you can delete a manually created host that is mapped, you must unmap it.

Syntax `delete host host`

Parameters *host*
The host ID or nickname.

Example Delete the manually created host MyHost:

```
# delete host MyHost
Info: Command completed successfully. - Host MyHost was deleted.
Success: Command completed successfully.
```

Try to delete the mapped host Host1:

```
# delete host Host1
Error: The specified host is mapped to one or more volumes so the host was not
deleted. - Host Host1 is mapped, so it cannot be deleted.
Error: Remove host failed - some host(s) may have been removed.
```

Try to delete the discovered host 500605B000320BB4:

```
# delete host 500605B000320BB4
Error: The specified host is discovered and can not be deleted. Disconnect the
host, restart the Storage Controllers, and retry. - Host 500605B000320BB4 was
discovered, so it cannot be deleted.
Error: Remove host failed - some host(s) may have been removed.
```

See also

- show host-maps
- show hosts

delete master-volume

Description Deletes a master volume.

NOTE: You must delete all snapshots that exist for the master volume before you can delete it.

Syntax `delete master-volume volume`

Parameters *volume*
Name or serial number of the master volume to delete. For syntax, see Command syntax.

Example Delete master volume MV1:

```
# delete master-volume MV1
Success: Command completed successfully. - The master volume was deleted.
```

See also
- `delete all-snapshots`
- `show master-volumes`

delete schedule

Description Deletes a task schedule.

Syntax `delete schedule` *`schedule`*

Parameters *`schedule`*
The schedule to delete.

Example Delete schedule `Sched1`:

```
# delete schedule Sched1
Success: Command completed successfully. - The schedule was deleted.
```

See also
- `show schedule-details`
- `show schedules`

delete snap-pool

Description Deletes a snap pool.

> **NOTE:** You must disassociate all master volumes from the snap pool before you can delete it.

Syntax `delete snap-pool` *`snap-pool`*

Parameters *`snap-pool`*
Name or serial number of the snap pool to delete. For syntax, see Command syntax.

Example Delete snap pool `SP1`:

```
# delete snap-pool SP1
Success: Command completed successfully. - The snap-pool was deleted.
```

See also
- `show master-volumes`
- `show snap-pools`

delete snapshot

Description Deletes a snapshot. All data uniquely associated with the snapshot is deleted and associated space in the snap pool is freed for use.

Syntax `delete snapshot [cleanup]` *`snapshot`*

Parameters `cleanup`
Optional. When a master volume's last snapshot is deleted, automatically convert the master volume to a standard volume and delete the snap pool.

`snapshot`
Name or serial number of the snapshot to delete. For syntax, see Command syntax.

Example Delete snapshot `SS1`:

```
# delete snapshot SS1
Success: Command completed successfully. - The snapshot was deleted.
```

See also
- `delete snapshot-write-data`
- `show snapshots`

delete snapshot-write-data

Description Deletes data written to a snapshot after it was created. Deleting this modified data reverts the snapshot to the state when it was first taken.

Syntax `delete snapshot-write-data` *`snapshot`*

Parameters *`snapshot`*
Name or serial number of the snapshot to delete modified data from. For syntax, see Command syntax.

Example Delete only modified data from snapshot `SS1`:

```
# delete snapshot-write-data SS1
Success: Command completed successfully. - Snapshot write data was deleted.
```

See also
- `delete snapshot`
- `show snapshots`

delete task

Description Deletes a task. If the task is scheduled, you must delete the schedule first.

Syntax `delete task` *`task`*

Parameters *`task`*
The task to delete.

Example Delete task `Task1`:

```
# delete task Task1
Success: Command completed successfully. - The task was deleted.
```

See also
- `delete schedule`
- `show schedule-details`
- `show schedules`
- `show task-details`
- `show tasks`

delete user

Description Deletes a user profile. You can delete any user except the default user `manage`.

Syntax `delete user [noprompt]` *`name`*

Parameters `noprompt`
Optional. Suppresses the confirmation prompt that requires a yes or no response.

`name`
The user to delete. Names are case sensitive.

Example Delete user `jsmith`:

```
# delete user jsmith
Are you sure you want to delete user jsmith? yes
Success: Command completed successfully. - The user was deleted.
```

Delete user `Kim` and suppress the confirmation prompt:

```
# delete user noprompt Kim
Success: Command completed successfully. - The user was deleted.
```

See also
- `show users`

delete vdisk

Description Deletes specified vdisks. This disassociates all disks that are assigned to the vdisks, and unmaps the vdisks' volumes.

△ **CAUTION:** Deleting a vdisk will delete all data on that vdisk.

NOTE: You cannot delete a vdisk if it contains a snap pool that is associated with a master volume on another vdisk.

Syntax `delete vdisk [prompt yes|no] vdisks`

Parameters `prompt yes|no`
Optional. Specifies an automatic response to the prompt that appears if a utility is running on the vdisk:

- `yes`: Stops the utility and enables the deletion to proceed
- `no`: Prevents the deletion from proceeding

If this parameter is omitted, you must manually reply to the prompt.

`vdisks`
Names or serial numbers of the vdisks to delete. For syntax, see Command syntax.

Example Delete vdisks `VD1 and VD2`:

```
# delete vdisk VD1,VD2
Please wait - vdisks are being deleted.
Info: Deleted vdisk VD1.
Success: Command completed successfully
Info: Deleted vdisk VD2.
Success: Command completed successfully
```

See also
- `show master-volumes`
- `show vdisks`

delete volume

Description Deletes specified volumes.

△ **CAUTION:** Deleting a volume will delete all data in that volume.

Syntax `delete volume volumes`

Parameters *`volumes`*
Names or serial numbers of the standard, master, snap-pool, or snapshot volumes to delete. For syntax, see Command syntax.

Example Delete volumes `V1 and V2`:

```
# delete volume V1,V2
Info: Command completed successfully. (V1) - Volume V1 was deleted.
Info: Command completed successfully. (V2) - Volume V2 was deleted.
Success: Command completed successfully.
```

See also
- `show volumes`

dequarantine

Description A previously fault-tolerant vdisk becomes quarantined when not all of its disks are detected after a restart or power cycle. Quarantine isolates the vdisk from host access, and prevents the storage system from making the vdisk critical and starting reconstruction when disks are "missing" for these reasons:

- Slow to spin up after system power-up
- Not properly seated in their slots
- In an powered-off enclosure
- Inserted from a different system and contains old metadata

The vdisk can be fully recovered if the missing disks can be restored. Make sure that no disks have been inadvertently removed and that no cables have been unplugged. Sometimes not all disks in the vdisk power up. Check that all enclosures have restarted after a power failure. If these problems are found and then fixed, the vdisk recovers and no data is lost.

The quarantined vdisk's disks are "write locked" and the vdisk is not available to hosts until the vdisk is removed from quarantine. The system waits indefinitely for the missing disks. If the disks are found, the system automatically removes the vdisk from quarantine. If the disks are never found because they have been removed or have failed, you must manually remove the vdisk from quarantine.

If the missing disks cannot be restored (for example, they failed), you can remove the vdisk from quarantine to restore operation in some cases. If you remove from quarantine a vdisk that is not missing too many disks, its status changes to critical. Then, if spares of the appropriate size are available, reconstruction begins.

NOTE: After you remove the vdisk from quarantine, make sure that a spare disk is available to let the vdisk reconstruct.

△ **CAUTION:** If the vdisk does not have enough disks to continue operation, when the vdisk is removed from quarantine it goes offline and its data cannot be recovered.

Syntax `dequarantine vdisk vdisk`

Parameters `vdisk vdisk`
Name or serial number of the vdisk to remove from quarantine. For syntax, see Command syntax.

Example After determining that vdisk `VD1` is quarantined, remove it from quarantine and re-check its status:

```
# show vdisks
Name ... Stat ...
-----------------
VD1  ... QTDN ... (Quarantined, Down)
-----------------

# dequarantine vdisk VD1
Success: Command completed successfully

# show vdisks
Name ... Stat ...
-----------------
VD1  ... FTDN ... (Fault Tolerant, Down)
```

See also
- show vdisks

exit

Description Log off and exit the CLI session.

Syntax `exit`

expand master-volume

See `expand volume`.

expand snap-pool

Description Expands a snap pool. Expansion is restricted to the space available on the vdisk containing the snap pool. If insufficient space is available for expansion on the vdisk, first expand the vdisk by using `expand vdisk`.

Syntax To expand by a specific size:

`expand snap-pool size` *`size`*`[B|KB|MB|GB|TB|KiB|MiB|GiB|TiB]` *`snap-pool`*

To expand to the maximum size:

`expand snap-pool size max` *`snap-pool`*

Parameters `size` *`size`*`[B|KB|MB|GB|TB|KiB|MiB|GiB|TiB]`
Specifies the size using the current base, as shown by `show cli-parameters`. The unit can be specified as follows:

- If base 2 is in use: `B` (bytes), `KiB` (kibibytes), `MiB` (mebibytes), or `GiB` (gibibytes)
- If base 10 is in use: `B` (bytes), `KB` (kilobytes), `MB` (megabytes), or `GB` (gigabytes)

If no unit is specified, the unit is 512-byte blocks.

`size max`
Expands the volume to fill available space on the vdisk.

`snap-pool`
Name or serial number of the snap pool to expand. For syntax, see Command syntax.

Example Expand snap pool `SP1` by 100 GB:

```
# expand snap-pool size 100GB SP1
Success: Snap-pool Expansion Started.
```

See also

- `show snap-pools`
- `show vdisks`

expand vdisk

Description Adds disks to a vdisk. The expansion capability for each supported RAID level is:

RAID level	Expansion capability	Maximum disks
NRAID	Cannot expand.	1
0, 3, 5, 6	Can add 1–4 disks at a time.	16
1	Cannot expand.	2
10	Can add 2 or 4 disks at a time.	16
50	Can expand the vdisk one RAID-5 sub-vdisk at a time. The added RAID-5 sub-vdisk must contain the same number of disks as each original sub-vdisk.	32

△ **CAUTION:** Vdisk expansion cannot be stopped and can take days to complete, depending on disk type, RAID level, and other factors.

Syntax `expand vdisk disks` *`disks vdisk`*

Parameters `disks` *`disks`*
IDs of the disks to add. For syntax, see Command syntax.

`vdisk`
Name or serial number of the vdisk to expand. For syntax, see Command syntax.

Example Expand vdisk `VD1` to include the disk having ID 11 in the first enclosure:

```
# expand vdisk disks 0.11 VD1
```

See also
- `show disks`
- `show vdisks`

expand volume

Description Expands a standard or master volume. Expansion is restricted to the space available on the vdisk containing the volume. If insufficient space is available for expansion on the vdisk, first expand the vdisk by using `expand vdisk`.

To expand a master volume:

1. Delete all of its snapshots by using `delete all-snapshots`.
2. Convert it to a standard volume by using `convert master-to-std`.
3. Expand the standard volume by using `expand volume`.
4. Convert the expanded volume to a master volume by using `convert std-to-master`.

Syntax To expand by a specific size:

`expand volume size` *`size`*`[B|KB|MB|GB|TB|KiB|MiB|GiB|TiB]` *`volume`*

To expand to the maximum size:

`expand volume size max` *`volume`*

Parameters `size` *`size`*`[B|KB|MB|GB|TB|KiB|MiB|GiB|TiB]`
Specifies the size using the current base, as shown by `show cli-parameters`. The unit can be specified as follows:

- If base 2 is in use: `B` (bytes), `KiB` (kibibytes), `MiB` (mebibytes), or `GiB` (gibibytes)
- If base 10 is in use: `B` (bytes), `KB` (kilobytes), `MB` (megabytes), or `GB` (gigabytes)

If no unit is specified, the unit is 512-byte blocks.

`size max`
Expands the volume to fill the available space on the vdisk.

`volume`
Name or serial number of the volume to expand. For syntax, see Command syntax.

Example Expand volume `V1` by 100 GB:

```
# expand volume size 100GB V1
Success: expanded volume V1
```

See also
- `expand master-volume`
- `expand vdisk`
- `show vdisks`
- `show volumes`

map volume

Description Maps a volume using settings that override the volume's default mapping.

When a volume is created, if no mapping settings are specified the volume is not mapped; otherwise, those settings become its default mapping, which specifies the controller host ports and access level that all connected hosts have to the volume, and the LUN presented to all hosts to identify the volume. The default mapping's LUN is known as the volume's *default LUN*.

The `map volume` command creates mappings with different settings for different hosts. Optionally, you can specify the LUN, ports, and access level for a mapping. A mapping can make a volume accessible to hosts, or inaccessible to hosts (known as *masking*). For example, assume a volume's default mapping allows read-only access using LUN 5. You can give one host read-write access using LUN 6, and you can give a second host no access to the volume.

Syntax

```
map volume
    [access read-write|rw|read-only|ro|no-access]
    [mapping ports.LUN]
    [lun LUN]
    [ports ports]
    [host host]
    volume
```

Parameters

`access read-write|rw|read-only|ro|no-access`
Optional. The access permission available to attached hosts: read-write (rw), read-only (ro), or no-access. When a volume is mapped with no-access, the volume is masked. If this parameter is omitted, access is set to read-write.

`mapping` *`ports.LUN`*
Optional. The ports and LUN to use for the mapping; any unspecified ports become unmapped. Ignored if access is set to `no-access`.

`lun` *`LUN`*
Optional. The LUN to use for the mapping. Ignored if access is set to `no-access`. If this parameter is omitted, the default LUN is presented. Do not use this parameter with the mapping parameter.

`ports` *`ports`*
Optional. The ports to use for the mapping; any unspecified ports become unmapped. Ignored if access is set to `no-access`. If this parameter is omitted, all ports are mapped. Use this parameter with the `lun` parameter but not with the mapping parameter.

`host` *`host`*
Optional. For FC and SAS, the host's nickname or 16-hex-digit WWPN. For iSCSI, the initiator's node name (typically the IQN) or nickname. For nickname syntax, see Command syntax. If this parameter is omitted, the mapping applies to all hosts that are not explicitly mapped.

`volume`
Name or serial number of the volume to map. For syntax, see Command syntax.

Example Map volume `v2` with read-only access for `Host1`, using port A1 and LUN 301. The volume will be unmapped for all other hosts:

```
# map volume access ro mapping a1.301 host Host1 v2
Success: Command completed successfully. - The volume was mapped successfully.
```

Also map volume `v2` with read-only access for `Host2`, using ports A1 and B1 and LUN 302. The volume will be unmapped for all hosts other than `Host1` and `Host2`:

```
# map volume access rw ports a1,b1 lun 302 host Host2 v2
Success: Command completed successfully. - The volume was mapped successfully.
```

See also
- `show host-maps`
- `show hosts`
- `show volume-maps`
- `show volumes`
- `unmap volume`

meta

Description For API use, shows all property metadata for objects. This includes data not shown in `brief` mode. The data is static and never changes.

Syntax `meta` *`basetypes`*

Parameters *`basetypes`*
A basetype or a list of basetypes separated by commas (with no spaces) to specify the objects to show metadata for. Basetypes are listed in XML API basetypes.

Example Show all metadata for objects returned by the `show disks` command:

```
# meta drives
```

ping

Description Tests communication with a remote host. The remote host is specified by IP address. Ping sends ICMP echo response packets and waits for replies.

Syntax `ping` *`host-address`* `[`*`count`*`]`

Parameters *`host-address`*
The remote host's IP address in dotted decimal form.

`count`
Optional. The number of packets to send. The default is 4 packets. Use a small count because the command cannot be interrupted.

Example Send two packets to the remote computer at 10.134.50.6:

```
# ping 10.134.50.6 2
Info: Pinging 10.134.50.6 with 2 packets.
Success: Command completed successfully. - The remote computer responded with 2
packets.
```

rescan

Description This command forces rediscovery of attached disks and enclosures. If both Storage Controllers are online this command also reassigns enclosure IDs based on controller A's enclosure cabling order. A manual rescan may be needed after system power-up to display enclosures in the proper order.

A manual rescan is not required to detect when disks are inserted or removed; the controllers do this automatically. When disks are inserted they are detected after a short delay, which allows the disks to spin up.

When you perform a manual rescan, it temporarily pauses all I/O processes, then resumes normal operation.

Syntax `rescan`

Example Scan for device changes and re-evaluate enclosure IDs:

```
# rescan
Success: Command completed successfully
```

reset host-link

Description Resets specified FC or SAS controller host ports (channels). For an FC host port configured to use FC-AL (loop) topology, a loop initialization primitive (LIP) is issued. For a SAS host port, a COMINIT/COMRESET sequence is issued.

Syntax `reset host-link ports ports`

Parameters `port ports`
A controller host port ID, a comma-separated list of IDs, a hyphenated range of IDs, or a combination of these. A port ID is a controller ID and port number, and is not case sensitive. Do not mix controller IDs in a range.

Example Reset the host link on port A1:

```
# reset host-link ports A1
Success: Command completed successfully. - Reset Host Link(s) on port(s) a1 from current controller.
```

See also
- show ports

reset snapshot

Description Deletes the data in a snapshot and resets it to the current data in the associated master volume. The snapshot's volume characteristics are not changed. The command prompts you to unmount the snapshot from the host operating system before performing the reset; leaving it mounted can cause data corruption.

△ **CAUTION:** All data represented by the snapshot as it exists prior to issuing this command is lost.

Syntax `reset snapshot [prompt yes|no] volume`

Parameters `prompt yes|no`
Optional. Specifies an automatic response to the unmount prompt that either enables the reset to proceed or prevents the reset from proceeding:

- `yes`: Enables the reset to proceed.
- `no`: Prevents the reset from proceeding.

If this parameter is omitted, you must reply to the prompt.

volume
Name or serial number of the snapshot to reset. For syntax, type Command syntax.

Example Reset snapshot `SS1`:

```
# reset snapshot SS1
Leaving the snapshot mounted during reset on any operating system can result in data corruption.
Is the snapshot unmounted from all operating systems? yes
Success: Command completed successfully. - The reset of a snapshot completed.
```

See also
- show snapshots

restart

Description Restarts the Storage Controller or Management Controller in a controller module.

If you restart a Storage Controller, it attempts to shut down with a proper failover sequence, which includes stopping all I/O operations and flushing the write cache to disk, and then the controller restarts. The Management Controller is not restarted so it can provide status information to external interfaces.

If you restart a Management Controller, communication with it is lost until it successfully restarts. If the restart fails, the partner MC remains active with full ownership of operations and configuration information.

△ **CAUTION:** If you restart both controller modules, you and users lose access to the system and its data until the restart is complete.

Syntax

```
restart sc|mc a|b|both [noprompt]
```

Parameters

`sc|mc`
The controller to restart:

- `sc`: Storage Controller
- `mc`: Management Controller

`a|b|both`
The controller module containing the controller to restart.

`noprompt`
Optional. Suppresses the confirmation prompt.

Example Restart the Management Controller in controller A, which you are logged in to:

```
# restart mc a
During the restart process you will briefly lose communication with the specified
management Controller(s).
Continue? yes
Info: Restarting Local MC A...
```

From controller A, restart the Storage Controller in controller B:

```
# restart sc b
Success: SC B restarted.
```

See also

- shutdown

restore defaults

Description **For use by or with direction from a service technician.**

Restores the manufacturer's default configuration to the controllers. When the command informs you that the configuration has been restored, you must restart the controllers for the changes to take effect. After restarting the controllers, hosts might not be able to access volumes until you re-map them.

△ **CAUTION:** This command changes how the system operates and might require some reconfiguration to restore host access to volumes.

Syntax

```
restore defaults [noprompt]
```

Parameters `noprompt`
Optional. Suppresses the confirmation prompt.

See also
- `restart`

rollback master-volume

Description Rolls back (reverts) the data on a master volume to the data that exists in a specified snapshot. You can choose whether to include modified write data from the snapshot in the rollback. You must unmount the master volume from the host operating system before using this command. The command will prompt you to ensure the master volume is unmounted before proceeding.

△ **CAUTION:** All data that differs between the master volume and the snapshot is lost. Create a snapshot of the master volume as it currently exists before performing a rollback.

Syntax
```
rollback master-volume
    snapshot snapshot
    [modifiedsnapshot yes|no]
    [prompt yes|no]
    master-volume
```

Parameters `snapshot` *`snapshot`*
Name or serial number of the snapshot containing the data to roll back to. For syntax, type Command syntax.

`modifiedsnapshot yes|no`
Optional. Specifies whether to include or exclude modified write data from the snapshot in the rollback.

- `yes`: Include modified snapshot.
- `no`: Exclude modified snapshot data.

If this parameter is omitted, modified snapshot data is excluded.

`prompt yes|no`
Optional. Specifies an automatic response to the unmount prompt that either enables the rollback to proceed or prevents the rollback from proceeding.

- `yes`: Enable the rollback to proceed.
- `no`: Prevent the rollback from proceeding.

If this parameter is omitted, you must reply to the prompt.

`master-volume`
Name or serial number of the master volume to roll back. For syntax, type Command syntax.

Example Roll back master volume `MV1` to snapshot `SS1`:

```
# rollback master-volume snapshot SS1 MV1
Leaving the master volume mounted when starting a rollback operation will result
in data corruption. The master volume must be unmounted prior to beginning the
rollback operation. The master volume can be remounted once the rollback has
started. Not unmounting the volume prior to beginning the rollback operation will
result in data corruption.
Is the master volume unmounted from all operating systems? yes
Success: Command completed successfully. - Rollback was started.
```

See also
- `show master-volumes`
- `show snapshots`

scrub vdisk

Description Analyzes specified vdisks to detect, report, and store information about disk defects. Vdisk-level errors reported include: hard errors, medium errors, and bad block replacements (BBRs). Disk-level errors reported include: metadata read errors, SMART events during scrub, bad blocks during scrub, and new disk defects during scrub.

For RAID 3, 5, 6, and 50, scrub checks all parity blocks to find data-parity mismatches. For RAID 1 and 10, scrub compares the primary and secondary disks to find data inconsistencies. For NRAID and RAID 0, scrub checks for media errors.

A scrub can last over an hour, depending on vdisk size, utility priority, and amount of I/O activity. However, a "foreground" scrub performed with this command is typically faster than a background scrub enabled with the `set job-parameters` command.

When the scrub is complete, the number of errors found is reported with event code 207 in the event log. You can use a vdisk while it is being scrubbed.

Syntax
```
scrub vdisk vdisks
```

Parameters *vdisks*
Names or serial numbers of the vdisks to scrub.

Example Scrub the disks in vdisk `vd1`:

```
# scrub vdisk vd1
Info: Scrub started on vdisk vd1
Success: Command completed successfully.
```

See also
- abort scrub
- show vdisks

set advanced-settings

Description For API use, sets advanced system configuration options.

Syntax
```
set advanced-settings
    [auto-write-back enabled|disabled|on|off]
    [background-scrub enabled|disabled|on|off]
    [compact-flash-failure enabled|disabled|on|off]
    [controller-failure enabled|disabled|on|off]
    [dynamic-spares enabled|disabled]
    [emp-poll-rate rate]
    [fan-failure enabled|disabled|on|off]
    [host-cache-control enabled|disabled]
    [missing-lun-response notready|illegal]
    [partner-firmware-upgrade enabled|disabled|on|off]
    [partner-notify enabled|disabled|on|off]
    [power-supply-failure enabled|disabled|on|off]
    [smart enabled|disabled|on|off]
    [super-cap-failure enabled|disabled|on|off]
    [sync-cache-mode immediate|flush]
    [temperature-exceeded enabled|disabled|on|off]
    [utility-priority low|medium|high]
```

Parameters `auto-write-back enabled|disabled|on|off`
Optional. Sets whether the cache mode automatically changes to write-back after the trigger condition is cleared.

`background-scrub enabled|disabled|on|off`
Optional. Sets whether disks are automatically checked for disk defects to ensure system health. The interval between background scrub finishing and starting again is 24 hours.

`compact-flash-failure enabled|disabled|on|off`
Optional. Sets whether the cache policy automatically changes to write-through when CompactFlash memory fails.

`controller-failure enabled|disabled|on|off`
Optional. Sets whether the cache policy automatically changes to write-through when a controller fails. The default is `disabled`.

`dynamic-spares enabled|disabled`
Optional. Sets whether the storage system will automatically designate a properly sized disk having Available status as a spare.

`emp-poll-rate` *`rate`*
Optional. Sets the interval at which the storage system polls the EC (EMP) for status changes, from 5–3600 seconds. The default is 5 seconds.

`fan-failure enabled|disabled|on|off`
Optional. Sets whether the cache policy automatically changes to write-through when a fan fails.

`host-cache-control enabled|disabled`
Optional. Sets whether the host can modify the cache setting. The default is `disabled`.

`missing-lun-response notready|illegal`
Optional. Sets the missing-LUN response which enables the host drivers to continue probing for LUNs until they reach the LUN to which they have access.

- `notready`: Sends a reply that there is a LUN where a gap has been created but that it's not ready. Sense data returned is sensekey = 2, code = 4, qualifier = 3. This option is the default.
- `illegal`: Sends a reply that there is a LUN but that the request is illegal. Sense data returned is sensekey = 5, code = 25h, qualifier = 0.

`partner-firmware-upgrade enabled|disabled|on|off`
Optional. Sets whether component firmware versions are monitored and will be automatically upgraded on the partner controller.

`partner-notify enabled|disabled|on|off`
Optional. Sets whether to notify the partner controller that a trigger condition occurred. Enable this option to have the partner also change to write-through mode for better data protection. Disable this option to allow the partner continue using its current caching mode for better performance. The default is `disabled`.

`power-supply-failure enabled|disabled|on|off`
Optional. Sets whether the cache policy automatically changes to write-through when a power supply fails.

`smart enabled|disabled|on|off`
Optional. Enables or disables Self-Monitoring Analysis and Reporting Technology in disk drives.

`super-cap-failure enabled|disabled|on|off`
Optional. Sets whether the cache policy automatically changes to write-through when cache backup power is not fully charged or fails.

`sync-cache-mode immediate|flush`
Optional. Sets the option that controls how the `SCSI SYNCHRONIZE CACHE` command is handled.

- `immediate`: Good status is returned immediately and cache content is unchanged. This option is the default.
- `flush`: Good status is returned only after all write-back data for the specified volume is flushed to disk.

`temperature-exceeded enabled|disabled|on|off`
Optional. Sets whether the system forces a controller shutdown if a temperature is detected that exceeds system threshold limits.

`utility-priority low|medium|high`
Optional. Sets the priority at which jobs (such as vdisk verification and reconstruction but not background scrub) run with respect to I/O operations competing for the system's processors: low, medium, or high.

Example Enable partner firmware upgrade:

```
# set advanced-settings partner-firmware-upgrade enabled
Info: Command completed successfully. - Parameter 'partner-firmware-upgrade' was set to 'enabled'.
Success: Command completed successfully. - The settings were changed successfully.
```

See also
- `show advanced-settings`

set auto-write-through-trigger

Description Sets the trigger conditions that cause the controller to change the cache policy from write-back to write-through. You can set multiple triggers. By default `super-cap-failure` and `auto-write-back` are enabled. Alias: `set awt`.

When the cache mode is changed, an event is logged.

Syntax

```
set auto-write-through-trigger
    [controller-failure enabled|disabled|on|off]
    [super-cap-failure enabled|disabled|on|off]
    [compact-flash-failure enabled|disabled|on|off]
    [power-supply-failure enabled|disabled|on|off]
    [fan-failure enabled|disabled|on|off]
    [temperature-exceeded enabled|disabled|on|off]
    [partner-notify enabled|disabled|on|off]
    [auto-write-back enabled|disabled|on|off]
```

Parameters

`controller-failure enabled|disabled|on|off`
If the cache policy is set to write-back, specify whether the policy automatically changes to write-through when a controller fails.

`super-cap-failure enabled|disabled|on|off`
If the cache policy is set to write-back, specify whether the policy automatically changes to write-through when cache backup power is not fully charged or fails.

`compact-flash-failure enabled|disabled|on|off`
If the cache policy is set to write-back, specify whether the policy automatically changes to write-through when CompactFlash memory fails.

`power-supply-failure enabled|disabled|on|off`
If the cache policy is set to write-back, specify whether the policy automatically changes to write-through when a power supply fails.

`fan-failure enabled|disabled|on|off`
If the cache policy is set to write-back, specify whether the policy automatically changes to write-through when a fan fails.

`temperature-exceeded enabled|disabled|on|off`
Specify whether to force a controller shutdown if a temperature is detected that exceeds system threshold limits.

`partner-notify enabled|disabled|on|off`
Specify whether to notify the partner controller that a trigger condition occurred. Enable this option to have the partner also change to write-through mode for better data protection. Disable this option to allow the partner continue using its current caching mode for better performance.

`auto-write-back enabled|disabled|on|off`
Specify whether the cache mode automatically changes to write-back after the trigger condition is cleared.

Example Enable the controller-failure trigger and disable the partner-notification trigger:

```
# set auto-write-through-trigger controller-failure enabled partner-notify
disabled
Success: Command completed successfully. - Auto-write-through-trigger parameters
were changed.
```

See also

- `show auto-write-through-trigger`
- `show events`

set awt

See `set auto-write-through-trigger`.

set cache-parameters

Description Sets a volume's cache options or the system's cache redundancy mode. Settings you can change include:

- Cache write policy
- Cache optimization mode
- Cache read-ahead size
- Cache redundancy mode

NOTE: Only change the read-ahead cache settings if you fully understand how the host operating system, application, and adapter move data so that you can adjust the settings accordingly. Be prepared to monitor system performance and adjust read-ahead size until you find the optimal size for your application.

Syntax To set cache options for a specified volume:

```
set cache-parameters
    [write-policy write-back|write-through]
    [optimization standard|super-sequential]
    [read-ahead-size disabled|default|maximum|64KB|128KB|256KB|512KB|1MB|2MB|4MB
    |8MB|16MB|32MB]
    volume
```

Parameters `write-policy write-back|write-through`
Optional. Sets the cache write policy to either:

- `write-back`: Write-back caching does not wait for data to be completely written to disk before signaling the host that the write is complete. This is the preferred setting for a fault-tolerant environment because it improves the performance of write operations and throughput.
- `write-through`: Write-through caching significantly impacts performance by waiting for data to be completely written to disk before signaling the host that the write is complete. Use this setting only when operating in an environment with low or no fault tolerance.

`optimization standard|super-sequential`
Optional. Sets the cache optimization mode to either:

- `standard`: Used for applications that read and write small files in random order, such as transaction-based and database update applications. Sets the cache block size to 32 KB.
- `super-sequential`: Used for applications that read and write large files in sequential order, such as video playback and multimedia post-production video and audio editing applications. Sets the cache block size to 128 KB.

`read-ahead-size`
Optional. Controls the use and size of read-ahead cache:

- `disable`: Turns off read-ahead cache.
- `default`: Sets one chunk for the first access in a sequential read and one stripe for all subsequent accesses.
- `maximum`: Allows the controller to dynamically calculate the maximum read-ahead cache size for the volume.
- `64KB`, `128KB`, `256KB`, `512KB`, `1MB`, `2MB`, `4MB`, `8MB`, `16MB`, `32MB`: Sets a specific cache size.

`volume`
Name or serial number of the volume to change. For syntax, type Command syntax.

Example Set the cache policy and cache optimization mode for volume `V1`:

```
# set cache-parameters optimization super-sequential read-ahead-size maximum V1
Success: Command completed successfully. - Cache parameters were changed
```

See also
- `show cache-parameters`
- `show volumes`

Cache write policy

The cache policy setting for each volume determines when cached data is written to the disks. The ability to hold data in cache while it is being written to disk can increase storage device speed during sequential reads.

- Write-back caching does not wait for data to be completely written to disk before signaling the host that the write is complete. This is the preferred setting for a fault-tolerant environment because it improves the performance of write operations and throughput. Write-back caching is enabled by default.
- Write-through caching significantly impacts performance by waiting for data to be completely written to disk before signaling the host that the write is complete. Use this setting only when operating in an environment with low or no fault tolerance.

You can configure the write policy to automatically change from write-back cache to write-through cache when certain environmental events occur, such as a fan failure. For details, see `set auto-write-through-trigger`.

Cache optimization mode

Before creating or modifying a volume, determine the appropriate cache optimization mode. The controller supports super-sequential optimization mode for sequential I/O and standard optimization mode for random I/O.

The cache optimization mode setting for each volume optimizes the cache block size used by the controller:

- For sequential optimization, the cache block size is 128 Kbyte.
- For random optimization, the cache block size is 32 Kbyte.

An appropriate cache block size improves performance when a particular application uses either large or small stripe sizes:

- Video playback, multimedia post-production audio and video editing, and similar applications read and write large files in sequential order.
- Transaction-based and database update applications read and write small files in random order.

Since the cache block size works in conjunction with the default stripe size set by the cache optimization mode for each volume you create, these default stripe sizes are consistent with the cache block size setting. You can, however, specify a different stripe size for any volume at the time you create it. For more information, see `create volume`.

Cache read-ahead size

You can optimize a volume for sequential reads or streaming data by changing the amount of data read in advance after two back-to-back reads are made. Read ahead is triggered by two back-to-back accesses to consecutive logical block address (LBA) ranges. Read ahead can be forward (that is, increasing LBAs) or reverse (that is, decreasing LBAs). Increasing the read-ahead size can greatly improve performance for multiple sequential read streams. However, increasing read-ahead size will likely decrease random read performance.

The default read-ahead size, which sets one chunk for the first access in a sequential read and one stripe for all subsequent accesses, works well for most users in most applications. The controllers treat volumes and mirrored vdisks (RAID 1) internally as if they have a stripe size of 64 Kbyte, even though they are not striped.

Cache redundancy mode

In the storage system's default operating mode, Active-Active, data for volumes configured to use write-back cache is automatically mirrored between the two controllers. Cache redundancy has a slight impact on performance but provides fault tolerance.

The operating mode applies per storage system, not per volume.

set chap-record

Description For iSCSI, changes an originator's CHAP record. This command is permitted whether or not CHAP is enabled.

Syntax

```
set chap-record
    name originator-name
    [secret originator-secret]
    [mutual-name recipient-name mutual-secret recipient-secret]
```

Parameters

`name originator-name`
The originator name; typically the originator's IQN.

`secret originator-secret`
The secret that the recipient uses to authenticate the originator. The secret is case sensitive and can include 12–16 bytes.

`mutual-name recipient-name`
Optional; for mutual CHAP only. The recipient name; typically the recipient's IQN. The name can have a maximum of 223 bytes, including 0–9, lowercase a–z, hyphen, colon, and period. A storage system's IQN is a controller host-port name without its `.a` or `.b` suffix; see the Port-Name field shown by the `show host-parameters` command. This parameter and `mutual-secret` must be set together.

`mutual-secret recipient-secret`
Optional; for mutual CHAP only. The secret that the originator uses to authenticate the recipient. The secret is case sensitive, can include 12–16 bytes, and must differ from the originator secret. A storage system's secret is shared by both controllers. This parameter and `mutual-name` must be set together.

Example For mutual CHAP, add a recipient name and secret to a CHAP record:

```
# set chap-record name iqn.1991-05.com.microsoft:myHost.domain mutual-name
iqn.1995-03.com.acme:01.storage.00c0ffd6000a mutual-secret ABCdef123456
```

See also

- `create chap-record`
- `delete chap-records`
- `show chap-records`
- `show host-parameters`
- `show iscsi-parameters`

set cli-parameters

Description Sets options that control CLI behavior. If you are accessing the CLI through the network port, settings apply to the current CLI session only. If you are accessing the CLI through the enclosure's CLI port, settings persist across sessions.

The base, locale, precision, temperature scale, timeout, and units settings are read from the user's profile, and can be overridden by using this command.

Syntax

```
set cli-parameters
    [base 2|10]
    [disk-addressing enclosure-slot]
    [console|api|api-embed]
    [brief enabled|disabled|on|off]
    [locale English|en]
    [pager enabled|disabled|on|off]
    [precision #]
    [storage-size-base 2|10]
    [storage-size-precision #]
    [storage-size-units auto|MB|GB|TB]
    [temperature-scale celsius|c|fahrenheit|f]
    [timeout #]
    [units auto|MB|GB|TB]
```

Parameters

`base 2|10`
Optional. Sets the base for entry and display of storage-space sizes:

- `2`: Sizes are shown as powers of 2, using 1024 as a divisor for each magnitude.
- `10`: Sizes are shown as powers of 10, using 1000 as a divisor for each magnitude.

Operating systems usually show volume size in base 2. Disk drives usually show size in base 10. Memory size is always shown in base 2.

`disk-addressing enclosure-slot`
Optional. Disks are shown, and must be specified, with the format *`enclosure-ID.disk-slot-number`*. This is the default. The `channel-id` option is not supported.

`console|api|api-embed`
Optional. Sets the output mode:

- `console`: Supports interactive use of the CLI by displaying command output in easily readable format. This mode automatically sizes fields according to content and adjusts content to window resizes. This is the default.
- `api`: Supports scripting by displaying command output in XML. All objects are displayed at the same level, related by `COMP` elements.
- `api-embed`: Alternate form of XML output which displays child objects embedded (indented) under parent objects. Enabling this option enables the `brief` option, by default.

`brief enabled|disabled|on|off`
Optional. When enabled, shows only the `name` and `key` properties of objects in API mode. Enabled by default.

`locale English|en`
Optional. The display language.

`pager enabled|on|disabled|off`
Optional. Specifies whether to halt output after each full screen to wait for keyboard input. Enabled by default. When using API mode, which is intended for scripting, disable paging.

`precision #`
Optional. Sets the number of decimal places (1–10) for display of storage-space sizes.

`storage-size-base 2|10`
Optional. Alias for `base`.

`storage-size-precision #`
Optional. Alias for `precision`.

`storage-size-units auto|MB|GB|TB`
Optional. Alias for `units`.

`temperature-scale celsius|c|fahrenheit|f`
Optional. Specifies to show temperatures in degrees Fahrenheit or Celsius.

`timeout #`
Optional. Sets the timeout value in seconds for the login session. Valid values are 30–9999, where 9999 means do not timeout. The default is 1800 seconds (30 minutes).

`units auto|MB|GB|TB`
Optional. Sets the unit for display of storage-space sizes. `auto` lets the system determine the proper unit for a size. Based on the `precision` setting, if the selected unit is too large to meaningfully display a size, the system uses a smaller unit for that size. For example, if the unit is set to `TB` and the precision is set to 1, the size 0.11709 TB is shown as 119.9 GB. Default is `auto`.

Example Set CLI parameters:

```
# set cli-parameters timeout 600 precision 2 units GB temperature-scale f
Success: Command completed successfully. - The settings were changed
successfully.
```

For scripting, display XML output in API mode with no paging:

```
# set cli-parameters api pager off
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<RESPONSE VERSION="W440" request="cli-parameters">
<OBJECT basetype="status" name="status" oid="1">
    <PROPERTY name="response-type" type="enumeration" size="12" draw="false"
sort="nosort" display-name="Response Type">Success</PROPERTY>
    <PROPERTY name="response-type-numeric" type="enumeration" size="12"
draw="false" sort="nosort" display-name="Response">0</PROPERTY>
    <PROPERTY name="response" type="string" size="180" draw="true" sort="nosort"
display-name="Response">Command completed successfully. - The settings were
changed successfully.</PROPERTY>
   <PROPERTY name="return-code" type="int32" size="5" draw="false" sort="nosort"
display-name="Return Code">0</PROPERTY>
    <PROPERTY name="component-id" type="string" size="80" draw="false"
sort="nosort" display-name="Component ID"></PROPERTY>
</OBJECT>
</RESPONSE>
```

For scripting, display brief XML output in API-embed mode with no paging:

```
# set cli-parameters api-embed pager off brief on
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<RESPONSE VERSION="W440" request="cli-parameters">
<OBJECT basetype="status" name="status" oid="1">
    <PROPERTY name="response-type">Success</PROPERTY>
    <PROPERTY name="response-type-numeric">0</PROPERTY>
   <PROPERTY name="response">Command completed successfully. - The settings were
changed successfully.</PROPERTY>
    <PROPERTY name="return-code">0</PROPERTY>
    <PROPERTY name="component-id"></PROPERTY>
</OBJECT>
</RESPONSE>
```

Set CLI to show output in console mode:

```
# set cli-parameters console
Success: Command completed successfully. - The settings were changed
successfully.
```

See also • `show cli-parameters`

set controller-date

Description Sets the date and time for each Management Controller and then updates the date and time for each Storage Controller.

Syntax
```
set controller-date
    jan|feb|mar|apr|may|jun|jul|aug|sep|oct|nov|dec
    day
    hh:mm:ss
    year
    time-zone
    [ntp enabled|disabled|on|off]
    [ntpaddress IP-address]
```

Parameters `jan|feb|mar|apr|may|jun|jul|aug|sep|oct|nov|dec`
The month.

day
The day number (1–31).

hh:*mm*:*ss*
The hour on a 24-hour clock (0–23), the minutes (0–59), and the seconds (0–59).

year
The year as a four-digit number.

time-zone
The system's time zone based on an offset from Universal Time (UT) in hours.

`ntp enabled|disabled|on|off`
Optional. Enables use of Network Time Protocol (NTP). If an NTP server is available, the controller's time is synchronized with the server. When enabling NTP you must specify all the parameters but only the `ntpaddress` and *time-zone* parameters must be set accurately; the other parameters are ignored. When NTP is enabled, a client task becomes active.

Example Set the controller date to 1:45 PM on September 22, 2006 in the U.S. Mountain time zone (GMT -07:00):

```
# set controller-date sep 22 13:45:0 2007 -7
Success: Command completed successfully
```

Enable NTP using a specified time-zone offset and NTP server address:

```
# set controller-date sep 22 13:45:0 2007 -7 ntp enabled ntpaddress 69.10.36.3
Success: Command completed successfully
```

See also
- show controller-date
- show ntp-status

set debug-log-parameters

Description **For use by or with direction from a service technician.**

Sets the types of debug messages to include in the Storage Controller debug log. If multiple types are specified, use spaces to separate them and enclose the list in double quotes.

Syntax `set debug-log-parameters` *`message-type`*`+|- [...]`

Parameters *`message-type`*`+|-`
One of the following message types, followed by a plus (+) to enable or a minus (-) to disable inclusion in the log:

- `awt`: Auto-write-through feature debug messages
- `bkcfg`: Internal configuration debug messages
- `cache`: Cache debug messages
- `capi`: Internal Configuration API debug messages
- `capi2`: Internal Configuration API tracing debug messages
- `disk`: Disk interface debug messages
- `dms`: Snapshot feature debug messages
- `emp`: Enclosure Management Processor debug messages
- `fo`: Failover/recovery debug messages
- `fruid`: FRU ID debug messages
- `host`: Host interface debug messages
- `ioa`: I/O interface driver debug messages (standard)
- `iob`: I/O interface driver debug messages (resource counts)
- `ioc`: I/O interface driver debug messages (upper layer, verbose)
- `iod`: I/O interface driver debug messages (lower layer, verbose)
- `mem`: Internal memory debug messages
- `misc`: Internal debug messages
- `msg`: Inter-controller message debug messages
- `mui`: Internal service interface debug messages
- `raid`: RAID debug messages
- `rcm`: Removable-component manager debug messages
- `res2s`: Internal debug messages

Example Include RAID and cache messages, exclude EMP messages, and leave other message types unchanged:

```
# set debug-log-parameters "raid+ cache+ emp-"
Success: Command completed successfully. - Debug-log parameters were changed.
```

See also

- `show debug-log-parameters`

set disk-parameters

Description Specifies a Self-Monitoring Analysis and Reporting Technology (SMART) setting for all disks in the storage system. Disks equipped with this technology can alert the controller of impending disk failures. Changes to the SMART setting take effect after a rescan or a controller restart.

Syntax
```
set disk-parameters smart enabled|disabled|on|off|detect-only
```

Parameters

`smart enabled|disabled|on|off|detect-only`
Enables or disables SMART monitoring for all disks in the storage system. Each disk added after detect-only is set retains its SMART setting. Enabled by default.

When SMART is enabled, the system checks for SMART events one minute after a restart and every five minutes thereafter. SMART events are recorded in the event log.

Example Enable SMART:

```
# set disk-parameters smart on
Success: Command completed successfully
```

See also
- show disk-parameters

set email-parameters

Description Sets SMTP parameters for event notification.

Syntax
```
set email-parameters
   server server
   domain domain
   email-list email-addresses
   notification-level none|info|warn|crit
   [sender sender]
```

Parameters

`server` *server*
The IP address of the SMTP mail server to use for the email messages.

`domain` *domain*
The domain name that, with the sender name, forms the "from" address for remote notification.

`email-list` *email-addresses*
Enter up to four comma-separated email addresses for recipients of event notifications.

`notification-level none|info|warn|crit`
The minimum severity for which the system should send notifications: Informational (`info`), Warning (`warn`), Critical (`crit`). The default is none, which disables email notification and clears the settings.

`sender` *sender*
Optional. The sender name that, with the domain name, forms the "from" address for remote notification.

Example Set the system to send an email from `RAIDsystem@mydomain.com` to `sysadmin@mydomain.com` when a Warning event occurs:

```
# set email-parameters server 10.1.1.10 domain mydomain.com email-list
sysadmin@mydomain.com notification-level warn sender RAIDsystem
Info: Set Email Address 1 to: sysadmin@mydomain.com
Info: Set Email Server Name to: 10.1.1.10
Info: Set Email Domain Name to: mydomain.com
Info: Set Email Notification Level to: warn
Info: Set Email Sender Name to: RAIDsystem
Success: Command completed successfully.
```

See also
- show email-parameters
- test

set enclosure

Description Sets an enclosure's name, location, rack number, and rack position. Set these parameters to values that help you identify and locate the enclosure. A value that contains a space must be enclosed in double quotes.

These values are used when user interfaces show enclosure-related data; for example, in output of the `show enclosures` command and in event-log entries related to enclosures.

Syntax

```
set enclosure
    [name new-name]
    [location location]
    [rack-number rack-number]
    [rack-position rack-position]
    enclosure-number
```

Parameters

`name new-name`
A new name for the enclosure. The name can include a maximum of 20 bytes, using characters except double quote or backslash.

`location location`
The location of the enclosure. The name can include a maximum of 20 bytes, using characters except double quote or backslash.

`rack-number rack-number`
The number of the rack containing the enclosure, from 0–255.

`rack-position rack-position`
The enclosure's position in the rack, from 0–255.

enclosure-number
The enclosure ID.

Example Set parameters for enclosure 1:

```
# set enclosure name Storage-5 location Lab rack-number 9 rack-position 3 1
Success: Command completed successfully.
```

See also

- `show enclosures`

set expander-fault-isolation

Description **For use by or with direction from a service technician.**

By default, the Expander Controller in each I/O module performs fault-isolation analysis of SAS expander PHY statistics. When one or more error counters for a specific PHY exceed the built-in thresholds, the PHY is disabled to maintain storage system operation.

While troubleshooting a storage system problem, a service technician can use this command to temporarily disable fault isolation for a specific Expander Controller in a specific enclosure.

NOTE: If fault isolation is disabled, be sure to re-enable it before placing the system back into service. Serious problems can result if fault isolation is disabled and a PHY failure occurs.

Syntax

```
set expander-fault-isolation
    wwn enclosure-wwn
    controller a|b|both
    enabled|disabled|on|off
```

Parameters

`wwn enclosure-wwn`
The WWPN of the enclosure containing the Expander Controller whose setting you want to change. To determine the WWPN, use `show enclosure-status`.

`controller a|b|both`
The I/O module containing the Expander Controller whose setting you want to change: A, B, or both.

`enabled|disabled|on|off`
Whether to enable or disable PHY fault isolation.

Example Disable PHY fault isolation for Expander Controller A in an enclosure:

```
# set expander-fault-isolation wwn 500C0FF00A408A3C controller a disabled
Success: Disabled expander fault isolation.
```

Re-enable PHY fault isolation for Expander Controller A in the same enclosure:

```
# set expander-fault-isolation wwn 500C0FF00A408A3C controller a enabled
Success: Enabled expander fault isolation.
```

See also
- `set expander-phy`
- `show enclosure-status`
- `show expander-status`

`set expander-phy`

Description **For use by or with direction from a service technician.**

Disables or enables a specific PHY.

Syntax

```
set expander-phy
   encl enclosure-ID
   controller a|b|both
   type phy-type
   phy phy-ID
   enabled|disabled|on|off
```

Parameters

`encl` *`enclosure-ID`*
The enclosure ID of the enclosure containing the PHY.

`controller a|b|both`
The I/O module containing the PHY to enable or disable: A, B, or both.

`type` *`phy-type`*
The PHY type, which can be one of the following values:

- `drive`
- `egress`
- `ingress`
- `sc-0` (ingress bus to the local SC)
- `sc-1` (ingress bus to the partner SC)

`phy` *`phy-ID`*
The logical PHY number.

`enabled|disabled|on|off`
Whether to enable or disable the specified PHY.

Example Disable the first egress PHY in controller A, and check the resulting status:

```
# set expander-phy encl 0 controller a type egress phy 0 disabled
Success: Command completed successfully. - Disabled PHY 0 on controller a in
enclosure 0. (PHY type: EGRESS)

# show expander-status
Encl   Ctlr Phy  Type         Status           Elem Status   Disabled  Reason
------------------------------------------------------------------------------
...
0      A    0    Egress       Disabled         Disabled      Disabled  0x8003
```

Enable the PHY for disk 5 in controller B, and check the resulting status:

```
# set expander-phy encl 0 controller b type drive phy 5 enabled
Success: Command completed successfully. - Enabled PHY 5 on controller b in
enclosure 0. (PHY type: DRIVE)

# show expander-status
Encl   Ctlr Phy  Type         Status           Elem Status   Disabled  Reason
--------------------------------------------------------------------------------
...
0      B    5    Drive        Enabled-Healthy  OK            Enabled
```

See also

- set expander-fault-isolation
- show enclosure-status
- show expander-status

set host-name

Description Changes a host's nickname.

Syntax

```
set host-name
    id host

    new-nickname
```

Parameters

`id host`
The ID or nickname of the host to rename.

new-nickname
A new nickname for the host. The name is case sensitive, cannot include a comma, double quote, or backslash, and can have a maximum of 15 bytes. A name that includes a space must be enclosed in double quotes.

Example Change a host's nickname to `MyHost`:

```
# set host-name id Host1 MyHost
Success: Command completed successfully. - The host was renamed.
```

See also

- show hosts

set host-parameters

Description Sets controller host port parameters for communication with attached hosts. The new settings take effect without restarting the controllers. There are parameters for FC and iSCSI ports but not for SAS ports.

Syntax

```
set host-parameters
    [controller a|b|both]
    [speed 1g|2g|4g|auto]
    [ports #|all]
    [fibre-connection-mode loop|point-to-point]
    [ip address]
    [netmask address]
    [gateway address]
    [prompt yes|no|y|n|expert]
    [noprompt]
    [fibre-loop-id values]
```

Parameters `controller a|b|both`
Optional. Specifies whether to apply the settings to all host ports in controller A, B, or both.

`speed 1g|2g|4g|auto`
Optional. Sets a forced link speed in Gbit/sec or lets the speed be auto-negotiated (`auto`). Use `auto` if the port is directly connected to a host or switch. Because a speed mismatch prevents communication between the port and host, set a speed only if you need to force the port to use a known speed for testing, or you need to specify a mutually supported speed for more than two FC devices connected in an arbitrated loop.

`ports #|all`
Optional. Specific host port numbers or all ports. For syntax, see Command syntax. If this parameter is omitted, all host ports on the specified controllers are affected.

`fibre-connection-mode loop|point-to-point`
Optional. For FC, sets the topology for the specified ports to either:

- `loop`: Fibre Channel-Arbitrated Loop (public or private)
- `point-to-point`: Fibre Channel point-to-point.

You must also specify the `controller` parameter.

`ip` *`address`*
Optional. For iSCSI, the port IP address in IPv4 format.

`netmask` *`address`*
Optional. For iSCSI, the port netmask address in IPv4 format.

`gateway` *`address`*
Optional. For iSCSI, the port gateway address in IPv4 format.

`prompt yes|no|y|n|expert`
Deprecated; use the `noprompt` parameter instead.

`noprompt`
Optional. Specifies an automatic response to the confirmation prompt that enables the operation to proceed. If this parameter is omitted, you must reply to the prompt.

`fibre-loop-id` *`values`*
Optional. For FC, specifies comma-separated loop ID values to request for host ports when controllers arbitrate during a LIP. Use this option if you want ports to have specific addresses, if your system checks addresses in reverse order (lowest address first), or if an application requires that specific IDs be assigned to recognize the controller. If the loop ID is changed for one port, the same ID is used for other ports in the same controller. If the `ports` parameter is specified, loop IDs are set based on the controllers that the ports are in. You cannot specify the same value for ports on different controllers.

- `soft` or 255: Soft target addressing enables the LIP to determine the loop ID. Use this setting if the loop ID is permitted to change after a LIP or power cycle.
- 0–125: Specify a hard target address if you do not want the loop ID to change after a LIP or power cycle. If the port cannot acquire the specified ID, it is assigned a soft target address.

You must restart affected controllers to make loop ID changes take effect.

Example On a system with FC ports, set the link speed to 2 Gbit/sec for ports A1 and B1:

```
# set host-parameters speed 2g ports a1,b1
WARNING: This change will take effect immediately. Changes may affect access to
data. Are you sure? yes
Success: Command completed successfully. - The host parameters were changed
successfully
```

On a system with FC ports, set controller A ports to request loop ID 14 and controller B ports to use soft target addressing, and suppress the confirmation prompt:

```
# set host-parameters fibre-loop-id 14,soft noprompt
Info: Changes to FC loop ids will not take effect until the affected Storage
Controller is restarted.
Success: Command completed successfully. - The host parameters were changed
successfully
```

On a system with FC ports, set the link speed to 4 Gbit/sec for ports A0, B0, B1; set controller A ports to request loop ID 5 and controller B ports to request loop ID 32; and suppress the confirmation prompt:

```
# set host-parameters speed 4g ports a0,b0,b1 fibre-loop-id 5,32 noprompt
Info: Changes to FC loop ids will not take effect until the affected Storage
Controller is restarted.
Success: Command completed successfully. - The host parameters were changed
successfully
```

On a system with iSCSI ports, for port A1 force the link speed to 1 Gbit/sec and change the IP address:

```
# set host-parameters speed 1g ip 10.134.50.6 port a1
WARNING: This change will take effect immediately. Changes may affect access to
data. Are you sure? yes
Success: Command completed successfully. - The host parameters were changed
successfully
```

See also

- `show host-parameters`

set iscsi-parameters

Description For iSCSI, changes system-wide iSCSI parameters.

Syntax

```
set iscsi-parameters
    [chap enabled|disabled|on|off]
    [jumbo-frame enabled|disabled|on|off]
    [speed auto|1g]
    [isns enabled|disabled|on|off]
    [isns-ip iSNS-IP]
    [isns-alt-ip iSNS-IP]
```

Parameters

`chap enabled|disabled|on|off`
Enables or disables use of Challenge Handshake Authentication Protocol. Disabled by default.

`jumbo-frame enabled|disabled|on|off`
Enables or disables support for jumbo frames. Disabled by default. A normal frame can contain 1500 bytes whereas a jumbo frame can contain a maximum of 9000 bytes for larger data transfers.

`speed auto|1g`
Sets the host port link speed either to `auto`, which allows the system to negotiate the proper speed, or to 1 Gbit/sec (`1g`). The default is `auto`.

`isns enabled|disabled|on|off`
Enables or disables registration with a specified Internet Storage Name Service server, which provides name-to-IP-address mapping. Disabled by default.

`isns-ip` *`iSNS-IP`*
Specifies the IP address of an iSNS server. The default address is all zeroes.

`isns-alt-ip` *`iSNS-IP`*
Optional. Specifies the IP address of an alternate iSNS server, which can be on a different subnet. The default address is all zeroes.

Example For a storage system whose host ports are connected to different subnets, enable CHAP, specify the IP address of the iSNS server on each subnet, and enable registration with either server:

```
# set iscsi-parameters chap enabled isns enabled isns-ip 10.10.10.93 isns-alt-ip
10.11.10.90
```

See also
- `show iscsi-parameters`

set job-parameters

Description Sets parameters for background scrub, partner firmware upgrade, and other jobs.

Syntax

```
set job-parameters
    [background-scrub enabled|disabled|on|off]
    [partner-firmware-upgrade enabled|disabled|on|off]
    [utility-priority low|medium|high]
```

Parameters

`background-scrub enabled|disabled|on|off`
Optional. Sets whether vdisks are checked for disk defects to ensure system health. The interval between background scrub finishing and starting again is 24 hours.

`partner-firmware-upgrade enabled|disabled|on|off`
Optional. Sets whether versions of firmware components are monitored and automatically upgraded on the partner controller.

`utility-priority low|medium|high`
Optional. Sets the priority at which jobs run with respect to I/O operations competing for the system's processors. This affects vdisk verification and reconstruction, but not background scrub.

Example Enable background scrubbing of vdisks and disable partner firmware upgrade:

```
# set job-parameters background-scrub on partner-firmware-upgrade off
Info: Command completed successfully. - Parameter 'background-scrub' was set to
'on'.
Info: Command completed successfully. - Parameter 'partner-firmware-upgrade' was
set to 'off'.
Success: Command completed successfully. - The settings were changed
successfully.
```

See also
- `show job-parameters`

set led

Description Changes the state of the Unit Identification (UID) LED on a specified disk or enclosure. LEDs are described in the setup guide.

Syntax To set a disk LED:

```
set led disk ID on|off
```

To set an enclosure LED:

```
set led enclosure ID on|off
```

Parameters

`disk ID`
The disk to locate. For syntax, see Command syntax.

`enclosure ID`
The enclosure to locate.

`on|off`
Specifies to set or unset the LED.

Example Identify disk 5 in the first enclosure:

```
# set led disk 0.5 on
Success: Command completed successfully. - Enabling identification LED for disk
0.5...
```

Stop identifying the first enclosure:

```
# set led enclosure 0 off
Success: Disabling identification LED for enclosure 0...
```

set network-parameters

Description Sets IP values for controller module network ports. IP values can be set dynamically using Dynamic Host Configuration Protocol (DHCP) for both controllers, or manually (statically) for each controller.

If DHCP is enabled, manually setting an IP value for either controller disables DHCP for both controllers.

Syntax To set both controllers' IP values dynamically:

```
set network-parameters dhcp
```

To set a controller's IP values manually:

```
set network-parameters
    [ip address]
    [netmask netmask]
    [gateway gateway]
    controller a|b
```

Parameters `dhcp`
Specifies to use DHCP to set both controllers' IP values.

`ip` *`address`*
Optional. An IP address for the port.

`netmask` *`netmask`*
Optional. An IP subnet mask for the port.

`gateway` *`gateway`*
Optional. A gateway IP address for the port.

`controller a|b`
Specifies whether to apply the settings to controller A or B.

Example Use DHCP to set network port IP values:

```
# set network-parameters dhcp
```

Manually set network port IP values for controller A (disabling DHCP for both controllers, if it was enabled):

```
# set network-parameters ip 192.168.0.10 netmask 255.255.255.0 gateway
192.168.0.1 controller a
Success: Network parameters have been changed
```

See also
- show network-parameters

set password

Description Sets a user's password for system interfaces (such as the CLI). A password can be entered as part of the command, or the command prompts you to enter and re-enter the new password.

Syntax `set password [password password] [user]`

Parameters `password password`
Optional. A new password for this user. A password is case sensitive, cannot include a comma, double quote, or backslash, and can have a maximum of 19 bytes.

user
Optional. The user name to set the password for. If this argument is omitted, this command affects the logged-in user's password.

Example Change the password of the default user, `monitor`:

```
# set password monitor
Info: Changing password for user monitor.
Enter new password:****
Re-enter new password:****
Info: Command completed successfully. - The password was set.
Success: Command completed successfully.
```

See also
- show users

set prompt

Description Sets the prompt for the current CLI session.

Syntax `set prompt prompt`

Parameters *prompt*
The new prompt, which can include any valid UTF-8 characters except backslash, double quote, and control characters, and can have a maximum of 16 bytes. A prompt that includes a space must be enclosed in double quotes.

Example Set the prompt to `CLI>` followed by a space, and start entering another command:

```
# set prompt "CLI> "
Success: Command completed successfully
CLI> set ...
```

Set a null prompt, and start entering another command:

```
# set prompt ""
Success: Command completed successfully
set ...
```

set protocols

Description Enables or disables management services and protocols.

Syntax

```
set protocols
    [capi enabled|disabled|on|off]
    [debug enabled|disabled|on|off]
    [ftp enabled|disabled|on|off]
    [http enabled|disabled|on|off]
    [https enabled|disabled|on|off]
    [service enabled|disabled|on|off]
    [ses enabled|disabled|on|off]
    [smis enabled|disabled|on|off]
    [snmp enabled|disabled|on|off]
    [ssh enabled|disabled|on|off]
    [telnet enabled|disabled|on|off]
```

Parameters

`capi enabled|disabled|on|off`
Optional. Enables or disables the in-band CAPI management interface.

`debug enabled|disabled|on|off`
Optional. Enables or disables the Telnet debug port.

`ftp enabled|disabled|on|off`
Optional. Enables or disables the expert interface for updating firmware.

`http enabled|disabled|on|off`
Optional. Enables or disables the standard RAIDar web server.

`https enabled|disabled|on|off`
Optional. Enables or disables the secure RAIDar web server.

`service enabled|disabled|on|off`
Optional. Enables or disables the Telnet service port.

`ses enabled|disabled|on|off`
Optional. Enables or disables the in-band SCSI Enclosure Management Services interface.

`smis enabled|disabled|on|off`
Optional. Enables or disables the Storage Management Initiative Specification interface.

`snmp enabled|disabled|on|off`
Optional. Enables or disables the Simple Network Management Protocol interface. Disabling this option disables all SNMP requests to the MIB but still allows SNMP traps to be generated. To configure SNMP traps use the `set snmp-parameters` command.

`ssh enabled|disabled|on|off`
Optional. Enables or disables the secure shell CLI.

`telnet enabled|disabled|on|off`
Optional. Enables or disables the standard CLI.

Example Disable unsecure HTTP connections and enable FTP:

```
# set protocols http disabled ftp enabled
Success: Command completed successfully.
```

See also

- `show protocols`

set snap-pool-policy

Description Sets the recovery policy that determines the action taken when a specified snap pool's error and critical threshold levels are reached. The policy for the warning threshold is preset to `notifyonly`. A snap pool's default error policy is `deleteoldestsnapshot` and default critical policy is `deletesnapshots`.

Syntax

```
set snap-pool-policy
    [error autoexpand|deleteoldestsnapshot|deletesnapshots|haltwrites|notifyonly
    |nochange]
    [critical deleteoldestsnapshot|deletesnapshots|haltwrites|nochange]
    [autoexpansionsize size[B|KB|MB|GB|TB|KiB|MiB|GiB|TiB]]
    snap-pool
```

Parameters

`error autoexpand|deleteoldestsnapshot|deletesnapshots|haltwrites|notifyonly|nochange`
Optional. The policy to invoke when the error threshold level of snap-pool usage is reached.

- `autoexpand`: Automatically expand the snap pool using the `autoexpansionsize` value.
- `deleteoldestsnapshot`: Delete the oldest snapshot.
- `deletesnapshots`: Delete all snapshots.
- `haltwrites`: Halt writes to the snap pool.
- `notifyonly`: Generates an event to notify the administrator.
- `nochange`: Take no action.

`critical deleteoldestsnapshot|deletesnapshots|haltwrites|nochange`
Optional. Specifies the policy to invoke when the critical threshold level of snap-pool usage is reached.

`autoexpansionsize size[B|KB|MB|GB|TB|KiB|MiB|GiB|TiB]`
The amount by which the snap pool will be automatically expanded when the threshold level is reached. The value uses the current base, as shown by `set cli-parameters`. The unit can be specified as follows:

- If base 2 is in use: `B` (bytes), `KiB` (kibibytes), `MiB` (mebibytes), or `GiB` (gibibytes)
- If base 10 is in use: `B` (bytes), `KB` (kilobytes), `MB` (megabytes), or `GB` (gigabytes)

If no unit is specified, the unit is 512-byte blocks.

snap-pool
Name or serial number of the snap pool to set the policy for. For syntax, see Command syntax.

Example Set snap pool `SP1` to automatically expand by 10 GB when its error threshold is reached:

```
# set snap-pool-policy error autoexpand autoexpansionsize 10GB SP1
Success: Command completed successfully. - The snap-pool policy was changed.
```

See also

- `set snap-pool-threshold`
- `show snap-pools`

set snap-pool-threshold

Description Sets the percentages of snap-pool space used that trigger the warning and error threshold policies. Three threshold levels are defined:

- Warning indicates that snap-pool space is moderately full. When exceeded, an event is generated to warn the administrator.
- Error indicates that the snap pool is getting full and unless corrective action is taken, snapshot data loss is probable. When exceeded, an event is generated to warn the administrator and the associated snap-pool policy is triggered.
- Critical indicates that the snap pool is nearly full and that data loss is imminent. When exceeded, an event is generate to alert the administrator and the associated snap-pool policy is triggered. This threshold is preset to 99%.

Syntax `set snap-pool-threshold [warning #%] [error #%] snap-pool`

Parameters `warning #%`
The percent of snap-pool space used that triggers the warning threshold policy. This value must be less than the error threshold value.

`error #%`
The percent of snap-pool space used that triggers the error threshold policy. This value must be less than 99%.

`snap-pool`
Name or serial number of the snap pool to set the threshold for. For syntax, see Command syntax.

Example Set the warning and error thresholds for snap pool `SP1`:

```
# set snap-pool-threshold warning 60% error 85% SP1
Success: Command completed successfully. - The snap-pool threshold was changed.
```

See also
- `set snap-pool-policy`
- `show snap-pools`

`set snmp-parameters`

Description Sets SNMP parameters for event notification. To enable or disable SNMP requests to the MIB use the `set protocols` command.

Syntax
```
set snmp-parameters
    [enable crit|warn|info|none]
    [add-trap-host address]
    [del-trap-host address]
    [trap-host-list trap-host-list]
    [read-community string]
    [write-community string]
```

Parameters `enable crit|warn|info|none`
Optional. Sets the level of trap notification:

- `crit`: Sends critical events only.
- `warn`: Sends all critical events and warnings.
- `info`: Sends all events.
- `none`: All events are excluded from trap notification and traps are disabled.

`add-trap-host address`
Optional. Specifies the IP address of a destination host to send traps to. Three trap hosts can be set.

`del-trap-host address`
Optional. Deletes a trap destination host.

`trap-host-list trap-host-list`
Optional. Replaces the current list.

`read-community string`
Optional. Sets an alphanumeric community string for read-only access.

`write-community string`
Optional. Sets an alphanumeric community string for write access.

Example Enable critical events only, specify a trap host, and set the community string for read-only access:

```
# set snmp-parameters enable crit add-trap-host 172.22.4.171 read-community
public
Success: Command completed successfully. - SNMP parameters were changed.
```

See also
- `set protocols`
- `show snmp-parameters`
- `test`

set spares

Description Creates or deletes spare disks for redundant (RAID 1, 3, 5, 6, 10, 50) vdisks.

A global spare is available to any redundant vdisk with the same disk type. The system can have eight global spares. Each must have enough capacity to replace the smallest disk in any existing vdisk.

A dedicated spare is assigned to a redundant vdisk with the same disk type. A vdisk can have four spares. Each must have enough capacity to replace the smallest disk in that vdisk.

△ **CAUTION:** Existing spares not specified in this command are deleted and return to being available disks.

Syntax `set spares disks disks|none [vdisk vdisk]`

Parameters `disks disks|none`
IDs of the disks to designate as spares, or `none` to delete all global spares if no vdisk is specified or all dedicated spares for a specified vdisk. For disk syntax, see Command syntax.

`vdisk vdisk`
Optional. Name or serial number of the vdisk to assign spares to. For syntax, see Command syntax. If this parameter is omitted, the disks will be global spares.

Example Designate disk 1.2 as a global spare:

```
# set spares disks 1.2
Info: Command completed successfully. (1.2) - Global spare disk 1.2 was set.
Success: Command completed successfully.
```

Designate disk 1.3 as a dedicated spare for vdisk `VD1`:

```
# set spares disks 1.3 vdisk VD1
Info: Command completed successfully. (1.3) - Vdisk spare disk 1.3 was set.
Success: Command completed successfully.
```

Try to designate a disk of one type as a spare for a vdisk using a different disk type:

```
# set spares disk 2.1 vdisk VD1
Error: The specified spare cannot be used. (2.1) - Disk 2.1 is not the same type
as the ones currently used by the vdisks.
Error: Command failed.
```

Delete all global spares:

```
# set spares disks none
Success: Command completed successfully.
```

Delete all dedicated spares for vdisk VD1:

```
# set spares disks none vdisk VD1
Success: Command completed successfully.
```

See also
- show disks
- show vdisks

set system

Description Sets the system's name, contact person, location, and description. Each value can include a maximum of 79 bytes, using characters except double quote or backslash. A value that contains a space must be enclosed in double quotes.

Syntax
```
set system [name value] [contact value] [location value] [info value]
```

Parameters

`name value`
A name for the system.

`contact value`
A contact person for the system.

`location value`
The location of the system.

`info value`
Other information about the system.

Example Set the system name to `Test` and the contact to `J. Doe`:

```
# set system name Test contact "J. Doe"
Success: Command completed successfully
```

See also
- `show system`

set user

Description Changes user preferences for the session or permanently. You cannot change the access level of user `manage`. To change a user's password, use `set password`.

Syntax
```
set user
    [base 2|10]
    [interfaces interfaces]
    [level monitor|manage]
    [locale English|en]
    [password password]
    [precision #]
    [session-preferences]
    [storage-size-base 2|10]
    [storage-size-precision #]
    [storage-size-units auto|MB|GB|TB]
    [temperature-scale celsius|c|fahrenheit|f]
    [timeout #]
    [type standard|advanced|diagnostic]
    [units auto|MB|GB|TB]
    [user-name]
```

Parameters

`base 2|10`
Optional. Sets the base for entry and display of storage-space sizes:

- `2`: Sizes are shown as powers of 2, using 1024 as a divisor for each magnitude.
- `10`: Sizes are shown as powers of 10, using 1000 as a divisor for each magnitude.

Operating systems usually show volume size in base 2. Disk drives usually show size in base 10. Memory size is always shown in base 2.

`interfaces` *values*
Optional. Specifies the interfaces that the user can access. Multiple values must be separated by commas and no spaces.

- `cli`: Command-line interface.
- `wbi`: Web-browser interface.
- `ftp`: File transfer protocol interface.
- `none`: No interfaces.

`level monitor|manage`
Optional.

- `monitor`: User can view but not change system settings.
- `manage`: User can view and change system settings.

`locale English|en`
Optional. The display language.

`password` *password*
Optional. A password is case sensitive, cannot include a comma, double quote, or backslash, and can have a maximum of 19 bytes. If this parameter is omitted, the command prompts you to enter and re-enter a password for the user.

`precision #`
Optional. Sets the number of decimal places (1–10) for display of storage-space sizes.

`session-preferences`
Optional. Specifies that the current CLI settings will become permanent.

`storage-size-base 2|10`
Optional. Alias for `base`.

`storage-size-precision #`
Optional. Alias for `precision`.

`storage-size-units auto|MB|GB|TB`
Optional. Alias for `units`.

`temperature-scale celsius|c|fahrenheit|f`
Optional. Specifies to use the Celsius scale or Fahrenheit scale for temperature values.

`timeout #`
Optional. Sets the timeout value in seconds for the login session. Valid values are 30–9999, where 9999 means do not timeout. The default is 1800 seconds (30 minutes).

`type standard|advanced|diagnostic`
Optional. Specifies the user's level of technical expertise, to control access to functions in RAIDar.

- `standard`: Enables access to standard administrative functions.
- `advanced`: Enables access to standard and advanced functions.
- `diagnostic`: Enables access to standard, advanced, and troubleshooting functions.

`units auto|MB|GB|TB`
Optional. Sets the unit for display of storage-space sizes. `auto` lets the system determine the proper unit for a size. Based on the `precision` setting, if the selected unit is too large to meaningfully display a size, the system uses a smaller unit for that size. For example, if the unit is set to `TB` and the precision is set to 1, the size 0.11709 TB is shown as 119.9 GB. Default is `auto`.

user-name
Optional. Specifies the user profile to change. Names are case sensitive.

Example Change a user's type and interface access:

```
# set user jsmith type advanced interfaces wbi,cli
Success: User-Type set to advanced.
```

See also
- set password
- show users

set vdisk

Description Changes parameters for a specified vdisk.

Syntax `set vdisk [name new-name] [owner a|b] vdisk`

Parameters `name new-name`
Optional. A new name for the vdisk. For syntax, see Command syntax.

`owner a|b`
Optional. The new owner: controller A or B.

△ **CAUTION:** Before changing the owning controller for a vdisk, you must stop host I/O to the vdisk's volumes. Volume mappings are not affected.

---

vdisk
Name or serial number of the vdisk to change. For syntax, see Command syntax.

Example Rename vdisk `VD1` and change its owner to controller A:

```
# set vdisk name VD2 owner a VD1
Success: Command completed successfully
```

See also
- show vdisks

set volume

Description Changes a volume's name.

Syntax `set volume name new-name volume`

Parameters `name new-name`
A new name for the volume. For syntax, see Command syntax.

volume
Name or serial number of the volume to change. For syntax, see Command syntax.

Example Rename volume `V1` to `V2`:

```
# set volume name V2 V1
Success: Command completed successfully. - The volume was set.
```

See also
- show host-maps
- show volumes
- show volume-maps

show advanced-settings

Description For API use, shows the settings for advanced system-configuration options.

Syntax `show advanced-settings`

Output `Background Scrub`
Shows whether disks are automatically checked for disk defects to ensure system health.

`Partner Firmware Upgrade`
Shows whether component firmware versions are monitored and will be automatically upgraded on the partner controller.

`Utility Priority`
Priority at which jobs (such as vdisk verification and reconstruction but not background scrub) run with respect to I/O operations competing for the system's processors: High, Medium, or Low.

`SMART`
Shows whether Self-Monitoring Analysis and Reporting Technology is enabled or disabled for all disks, or is set to detect-only, which specifies that each new drive inserted in the system retains its current SMART setting.

`Dynamic Spare Configuration`
Shows whether the storage system will automatically designate a properly sized drive as a spare.

`Enclosure Polling Rate`
Shows the interval at which the storage system polls the EC (EMP) for status changes.

`Host Control of Caching`
Shows whether host control of write-back cache is enabled or disabled. When disabled, hosts cannot use the `SCSI MODE SELECT` command to change the storage system's cache setting.

`Sync Cache Mode`
Shows how the `SCSI SYNCHRONIZE CACHE` command is handled:

- `Immediate` – Good status is returned immediately and cache content is unchanged.
- `Flush To Disk` – Good status is returned only after all write-back data for the specified volume is flushed to disk.

`Missing LUN Response`
Missing LUN Response enables the host drivers to continue probing for LUNs until they reach the LUN to which they have access.

- `Not Ready` – Sends a reply that there is a LUN where a gap has been created but that its "not ready." Sense data returned is sensekey = 2, code = 4, qualifier = 3.
- `Illegal Request` – Sends a reply that there is a LUN but that the request is "illegal." Sense data returned is sensekey = 5, code = 25h, qualifier = 0.

`Controller Failure`
Shows whether the cache policy automatically changes to write-through when a controller fails.

`SuperCap Failure`
Shows whether the cache policy automatically changes to write-through when cache backup power is not fully charged or fails.

`CompactFlash Failure`
Shows whether the cache policy automatically changes to write-through when CompactFlash memory fails.

`Power Supply Failure`
Shows whether the cache policy automatically changes to write-through when a power supply fails.

`Fan Failure`
Shows whether the cache policy automatically changes to write-through when a fan fails.

`Temperature Exceeded`
Shows whether the system forces a controller shutdown when a temperature is detected that exceeds system threshold limits.

`Partner Notify`
Shows whether the partner controller is notified when a trigger condition occurs.

`Auto Write Back`
Shows whether the cache mode automatically changes to write-back after the trigger condition is cleared.

Example Show advanced system-configuration settings:

```
# show advanced-settings
Background Scrub: enabled
Partner Firmware Upgrade: disabled
Utility Priority: High
SMART: Enabled
Dynamic Spare Configuration: Disabled
Enclosure Polling Rate: 5
Host Control of Caching: enabled
Sync Cache Mode: Immediate
Missing LUN Response: Not Ready
Controller Failure: Disabled
SuperCap Failure: Enabled
CompactFlash Failure: Enabled
Power Supply Failure: Disabled
Fan Failure: Disabled
Temperature Exceeded: Disabled
Partner Notify: Disabled
Auto Write Back: Enabled
```

See also • `set advanced-settings`

show auto-write-through-trigger

Description Shows the system's write-through trigger settings. When a trigger condition occurs and the trigger is enabled, the RAID controller cache mode changes from write-back to write-through. Alias: `show awt`.

Syntax `show auto-write-through-trigger`

Example Show the system's auto-write-through trigger settings:

```
# show auto-write-through-trigger
Auto Write-Through Triggers
---------------------------
Controller Failure: Disabled
SuperCap Failure: Enabled
CompactFlash Failure: Enabled
Power Supply Failure: Disabled
Fan Failure: Disabled
Temperature Exceeded: Disabled
Partner Notify: Disabled
Auto Write Back: Enabled
```

See also • `set auto-write-through-trigger`

show awt

See `show auto-write-through-trigger`.

show cache-parameters

Description Shows cache settings and status for the system and optionally for a volume.

Syntax `show cache-parameters [volume]`

Parameters *volume*
Optional. Name or serial number of the volume to show settings for. For syntax, see Command syntax. If this parameter is not specified, only system-wide settings are shown.

Output System/controller cache parameters:

`Operation Mode`
The operation mode, also called the redundancy mode.

- `Active-Active ULP`: Both controllers are active using ULP (Unified LUN Provisioning).
- `Single-Controller`: There is only a single controller in the enclosure.

`Write Back Status`
Shows whether write-back caching is enabled for the controller

`CompactFlash Status`
Shows whether the controller's CompactFlash card is installed

`Cache Flush`
Shows whether cache flush is enabled for the controller

Volume cache parameters:

`Serial Number`
If a volume is specified, its serial number

`Name`
If a volume is specified, its name

`Cache Write Policy`
If a volume is specified, its cache policy, either write-back or write-through

`Cache Optimization`
If a volume is specified, its cache optimization mode, either standard (random) or super-sequential

`Read Ahead Size`
If a volume is specified, its read-ahead cache setting

Example Show the cache parameters for the system and for volume `V1`:

```
# show cache-parameters V1
System Cache Parameters
-----------------------
Operation Mode: Active-Active ULP

  Controller A Cache Parameters
  -----------------------------
  Write Back Status: Enabled
  CompactFlash Status: Installed
  Cache Flush: Enabled

  Controller B Cache Parameters
  -----------------------------
  Write Back Status: Enabled
  CompactFlash Status: Installed
  Cache Flush: Enabled

Volume Cache Parameters
-----------------------
Serial Number: 00c0ff0a906b0000dcaa834701000000
Name: V1
Cache Write Policy: write-back
Cache Optimization: standard
Read Ahead Size: Default
```

See also

- `set cache-parameters`
- `show volumes`

show chap-records

Description For iSCSI, shows all CHAP records or the record for a specific originator. This command is permitted whether or not CHAP is enabled.

Syntax `show chap-records [name originator-name]`

Parameters `name originator-name`
Optional. The originator name; typically the originator's IQN. If this parameter is omitted, the command displays CHAP records for all originators.

Output `Name`
Originator name

`Secret`
Secret that the recipient uses to authenticate the originator

`Mutual CHAP Name`
For mutual CHAP, the recipient name

`Mutual CHAP Secret`
For mutual CHAP, the secret that the originator uses to authenticate the recipient

Example Show the CHAP record for a specific host initiator:

```
# show chap-records name iqn.1991-05.com.microsoft:myHost.domain
CHAP Record(s)
--------------
Name: iqn.1991-05.com.microsoft:myHost.domain
Secret: 123456abcDEF
Mutual CHAP Name: iqn.1995-03.com.acme:01.storage.00c0ffd6000a
Mutual CHAP Secret: ABCdef123456
```

See also
- `create chap-record`
- `delete chap-records`
- `set chap-record`
- `show host-parameters`

show cli-parameters

Description Shows the current CLI session preferences.

Syntax `show cli-parameters`

Output `Timeout`
Timeout value in seconds for the login session.

`Output Format`

- `console`: Output is shown in human-readable format.
- `api`: Output is shown in XML format.

`Base`

- `2`: Storage sizes are entered and shown in base 2.
- `10`: Storage sizes are entered and shown in base 10.

`Pager`

- `enabled`: Output halts after each full screen until a key is pressed or all output is shown.
- `disabled`: All output is shown.

`Disk Mode`

`enclosure-slot`: Disks are shown, and must be specified, with the format *`enclosure-ID.disk-slot-number`*. For example, the second disk in the first enclosure has address 0.1.

`Locale`
Display language.

`Precision`
Number of decimal places shown for storage sizes.

`Units`

- `auto`: Storage sizes are shown in units determined by the system.
- `MB`: Storage sizes are shown in megabytes.
- `GB`: Storage sizes are shown in gigabytes.
- `TB`: Storage sizes shown in terabytes.

Based on the precision setting, if the selected unit is too large to meaningfully display a size, the system uses a smaller unit for that size.

`Temperature Scale`

- `Fahrenheit`: Temperatures are shown in degrees Fahrenheit.
- `Celsius`: Temperatures are shown in degrees Celsius.

Example Show current CLI settings:

```
# show cli-parameters
CLI Parameters
--------------
Timeout: 1800
Output Format: console
Brief Mode: disabled
Base: 10
Pager: enabled
Disk Mode: enclosure-slot
Locale: English
Precision: 1
Units: Auto
Temperature Scale: Celsius
```

See also
- `set cli-parameters`

show configuration

Description Shows system configuration information.

Syntax `show configuration`

Output
- System information from `show system`
- Controller information from `show controllers`
- Controller software and hardware version information from `versions`
- Host and expansion port information from `show ports`
- Disk information from `show disks`
- Disk information by enclosure from `show disks` with the `encl` option
- Vdisk information from `show vdisks`
- Enclosure status information, including SCSI Enclosure Services (SES) data from `show enclosure-status`
- Field-replaceable unit (FRU) information from `show frus`

show controller-date

Description Shows the system's current date and time.

Syntax `show controller-date`

Output `Controller Date`
Date and time in the format *yyyy-mm-dd hh:mm:ss*, where *hh* is the hour on a 24-hour clock

`Time-Zone Offset`
The system's time zone, shown as an offset in hours from Universal Time (UT)

Example Show the system date and time:

```
# show controller-date
Controller Date: 2008-06-09 10:58:04
Time-Zone Offset: -6
```

See also
- `set controller-date`
- `show ntp-status`

show controllers

Description Shows information about each controller in the storage system.

Syntax `show controllers`

Output `Controller ID`

`Serial Number`

`Hardware Version`

`CPLD Version`

`MAC Address`

`WWNN`

`IP Address`

`IP Subnet Mask`

`IP Gateway`

`Disks`
Number connected to this controller

`Vdisks`
Number owned by this controller

`Cache Memory Size (MB)`

`Host Ports`
Number of host ports

`Disk Channels`
Number of expansion ports

`Disk Bus Type`

`Status`
`Running`, `Down`, `Not Installed`, or `Unknown`

`Failed Over`
`No` or `Yes`

`Fail Over Reason`
If `Failed Over` is `Yes`, a reason for the failover appears; otherwise, `Not applicable` appears

Example Show information about a system with a single FC controller:

```
# show controllers
Controllers
-----------
Controller ID: A
Serial Number: SN
Hardware Version: 25
CPLD Version: 8
MAC Address: 00:C0:FF:27:50:24
WWNN: 500C0FF000013000
IP Address: 10.134.11.100
IP Subnet Mask: 255.255.0.0
IP Gateway: 10.134.0.1
Disks: 17
Vdisks: 2
Cache Memory Size (MB): 1024
Host Ports: 4
Disk Channels: 2
Disk Bus Type: SAS
Status: Running
Failed Over: No
Fail Over Reason: Not applicable
```

See also
- show configuration
- show frus

show debug-log-parameters

Description **For use by or with direction from a service technician.**

Shows which debug message types are enabled (on) or disabled (off) for inclusion in the Storage Controller debug log.

Syntax `show debug-log-parameters`

Example Show debug log parameters:

```
# show debug-log-parameters
Host: on
Disk: on
mem: off
...
```

See also
- set debug-log-parameters

show disk-parameters

Description Shows disk settings.

Syntax `show disk-parameters`

Output `SMART`
Shows whether SMART is enabled or disabled for all disks, or is set to `detect-only`, which specifies that each new disk inserted in the system retains its current SMART setting.

`Drive Write Back Cache`
Shows whether write-back cache for each disk in or added to the system will be enabled or disabled or a rescan or controller restart. `Don't Modify` means that each disk's current setting will not be modified.

`Timeout Retry Maximum`
Maximum number of times a timed-out I/O operation can be retried before the operation is failed.

`Attempt Timeout`
Number of seconds before an I/O operation is aborted and possibly retried.

`Overall Timeout`
Total time in seconds before an I/O operation is failed regardless of the `Attempt Timeout` and `Number of Retries` settings.

Example Show disk settings:

```
# show disk-parameters
Disk Parameters
-------------
SMART: Enabled
Drive Write Back Cache: Disabled
Timeout Retry Maximum: 3
Attempt Timeout: 8
Overall Timeout: 105
```

See also
- `set disk-parameters`

show disks

Description Shows information about all disks in the storage system. If no parameter is specified, the command shows information for all disks.

Syntax `show disks [disks|free|all|encl] [vdisk vdisk]`

Parameters *`disks`*`|free|all|encl`
Optional. Specifies the disks to report information about:

- *`disks`*: IDs of the disks to show information about. For syntax, see Command syntax.
- `free`: Shows information about all disks that are available.
- `all`: Shows information about all installed disks.
- `encl`: Shows information about all disks by enclosure. This option shows different fields than the other options and shows each disk slot, whether it contains a disk or not.

`vdisk` *`vdisk`*
Shows information for disks in the specified vdisk. For syntax, see Command syntax.

Output **Without the `encl` option:**

`Location`
Disk's enclosure and slot number

`Serial Number`
Disk serial number

`Vendor`
Disk vendor

`Rev`
Firmware revision number

`How Used`

- `AVAIL`: Available
- `GLOBAL SP`: Global spare
- `LEFTOVR`: Leftover
- `VDISK`: Used in a vdisk
- `VDISK SP`: Spare assigned to a vdisk

Any jobs running on the disk or its vdisk follow the state value:

- `EXPD`: The vdisk is being expanded
- `INIT`: The vdisk is being initialized
- `RCON`: The vdisk is being reconstructed
- `VRFY`: The vdisk is being verified
- `VRSC`: The vdisk is being scrubbed

`Type`

- SAS: Dual-port SAS
- SAS-S: Single-port SAS
- SATA: Dual-port SATA
- SATA-S: Single-port SATA

`Size`
Disk capacity

`Rate(Gb/s)`
Data transfer speed in Gbit/second

`SP`
Shows which controller a single-ported disk is connected to

`Status`
Disk status: Up, Warning, Error, Missing, Unknown, or Disconnected

With the `encl` option:

`Status`
Disk status: Up, Warning, Error, Missing, Unknown, or Disconnected

`Encl`
Enclosure number where the disk is located

`Slot`
Slot number in the enclosure where the disk is located

`Vendor`
Disk vendor

`Model`
Disk model number

`Serial Number`
Disk serial number

`Size`
Disk size

Example Show information for all disks:

```
# show disks
Location  Serial Number  Vendor  Rev  How Used    Type   Size    Rate(Gb/s) SP
  Status
-------------------------------------------------------------------------------
0.0       SN             vendor  rev  VDISK VRFY  SAS    146.8GB 3.0
  OK
0.1       SN             vendor  rev  AVAIL       SATA-S 120.0GB 1.5        A
  OK
0.2       SN             vendor  rev  LEFTOVR     SAS    146.8GB 3.0
  OK
...
1.0       SN             vendor  rev  GLOBAL SP   SAS    146.8GB 3.0
  OK
...
-------------------------------------------------------------------------------
```

Show information for all disks, by enclosure:

```
# show disks encl
Status       Encl Slot Vendor  Model           Serial Number            Size
-------------------------------------------------------------------------------
OK           0    0    vendor  model           SN                       146.8GB
...
Missing      0    2    N/A     N/A             N/A                      N/A
...
-------------------------------------------------------------------------------
```

See also • show vdisks

show email-parameters

Description Shows SMTP parameters for event notification.

Syntax `show email-parameters`

Example Show settings for email notification of events:

```
# show email-parameters
Email Parameters (SMTP)
-----------------------
Email Notification: enabled
Email Notify Filter: warn
Email Address 1: sysadmin@mydomain.com
Email Address 2:
Email Address 3:
Email Address 4:
Email Server: 10.1.1.10
Email Domain: mydomain.com
Email Sender: RAIDsystem
```

See also • set email-parameters

show enclosure-status

Description Shows the status of system enclosures and their components. For each attached enclosure, the command shows general SCSI Enclosure Services (SES) information followed by component-specific information.

Syntax `show enclosure-status`

Output **General SES fields:**

`Chassis`
Chassis serial number

`Vendor`
Enclosure vendor name

`Product ID`
Product model identifier

`CPLD`
Complex Programmable Logic Device version

`EMP # BUS:ID Rev`
Address and firmware revision of the Enclosure Management Processor in each controller's Expander Controller

`WWPN`
World wide port name of the SES device reporting the enclosure status

`Status`
Overall status of the enclosure

`Health`
Overall health of the enclosure: `OK`, `Degraded`, `Fault`, or `Unknown`

Enclosure Component Status fields:

`Type`
Component type:

- `FAN`: Cooling fan unit
- `PSU`: Power supply unit
- `Temp`: Temperature sensor
- `Voltage`: Voltage sensor
- `Disk`: Disk drive module

`#`
Unit ID

`Status`
Component status:

- `Absent`: The component is not present.
- `Fault`: The component or at least one subcomponent has failed.
- `Degraded`: The component or at least one subcomponent is not working normally.
- `OK`: The component and any subcomponents are working normally. Temperature status `OK` indicates that the sensor is working properly, not that the temperature is within an acceptable range.
- `N/A`: Status is not available.

`FRU P/N`
Part number of the field-replaceable unit (FRU) that contains the component

`FRU S/N`
Serial number of the FRU that contains the component

`Add'l Data`
Additional data such as fan speed (RPM), temperature (Celsius), voltage, or slot address

Example Show enclosure status:

```
# show enclosure-status
Chassis     Vendor    Product ID  CPLD  EMP A BUS:ID Rev  EMP B BUS:ID Rev
  WWPN              Status  Health
-------------------------------------------------------------------------
3CL816C013  vendor    product     8     0:31 1022         1:31 1022
  500C0FF00000133C  OK      OK
-------------------------------------------------------------------------

Type     #  Status  FRU P/N  FRU S/N  Add'l Data
-------------------------------------------------------------------------
FAN      01 OK      PN       SN       speed=4090
FAN      02 OK      PN       SN       speed=4170
PSU      01 OK      PN       SN       --
PSU      02 OK      PN       SN       --
Temp     01 OK      PN       SN       temp=37
...
Temp     04 OK      PN       SN       temp=33
Voltage  01 OK      PN       SN       voltage=11.86V
...
Voltage  10 OK      PN       SN       voltage=3.49V
Disk     01 OK      PN       SN       addr=0
Disk     02 OK      PN       SN       addr=1
...
-------------------------------------------------------------------------
```

See also • `show enclosures`

show enclosures

Description Full detail available in API output only. Shows information about the enclosures in the storage system.

Syntax `show enclosures`

Output `Encl`
Enclosure ID

`Encl WWN`
Enclosure WWN

`Name`
Enclosure name

`Location`
Enclosure location; blank if not set

`Rack`
Number of the rack containing the enclosure

`Pos`
Position of the enclosure in the rack

`Vendor`
Enclosure vendor

`Model`
Enclosure model

`EMP # BUS:ID Rev`
Address and firmware revision of the Enclosure Management Processor in each controller's Expander Controller

Example Show information about the enclosures:

```
# show enclosures
Encl Encl WWN          Name         Location   Rack Pos  Vendor    Model
  EMP A BUS:ID Rev EMP B BUS:ID Rev
-------------------------------------------------------------------------
0    WWN               Controller  Lab        0    4    vendor    model
  0:31 1030        1:31 1030
-------------------------------------------------------------------------
```

See also
- `set enclosure`
- `show enclosure-status`

show events

Description Shows events for an enclosure, including events from each Management Controller and each Storage Controller. A separate set of event numbers is maintained for each controller. Each event number is prefixed with a letter identifying the controller that logged the event.

Events are listed from newest to oldest, based on a timestamp with one-second granularity; therefore the event log sequence matches the actual event sequence within about one second.

If SNMP is configured, events can be sent to SNMP traps.

For event descriptions and recommended actions, see the R/Evolution event descriptions document.

Syntax To show a certain number of events:

`show events [last #] [a|b|both|error]`

To show events by date:

`show events [from date] [to date] [a|b|both|error]`

To show events by ID:

`show events [from-event ID] [to-event ID] [a|b|both|error]`

Parameters `last #`
Optional. Shows the latest specified number of events. If this parameter is omitted, all events are shown.

`from date`
Optional. Shows events including and after the specified date and time. Use the format *MMDDYYhhmmss*, where *hh* is the hour on a 24-hour clock. For example, 043008235900 represents Apr 30 2008 at 11:59:00 p.m. This parameter can be used with either `to` parameter.

`to date`
Optional. Shows events before and including the specified date and time. Use the format *MMDDYYhhmmss*, where *hh* is the hour on a 24-hour clock. For example, 043008235900 represents Apr 30 2008 at 11:59:00 p.m. This parameter can be used with either `from` parameter.

`from-event ID`
Optional. Shows events including and after the specified event ID. If this number is smaller than the ID of the oldest event, events are shown from the oldest available event. This parameter can be used with either `to` parameter.

`to-event ID`
Optional. Shows events before and including the specified event ID. If this number is larger than the ID of the oldest event, events are shown up to the latest event. This parameter can be used with either `from` parameter.

`a|b|both|error`
Optional. Specifies to filter the event listing:

- `a`: Shows events from controller A only
- `b`: Shows events from controller B only
- `error`: Shows all critical and warning events but no informational events

Output

- Day, date, time, and year when the event was logged
- Event code identifying the type of event to help service technicians diagnose problems; for example, `[181]`
- Event ID prefixed by `A` or `B`, indicating which controller logged the event; for example, `#A123`
- Model, serial number, and ID of the controller module that logged the event
- Severity: `CRITICAL` events can affect data integrity or system stability; `WARNING` events do not affect data integrity; `INFORMATIONAL` events show state or configuration changes
- Event-specific message giving details about the event

Example Show the last two events:

```
# show events last 2
```

Show the last three warning and critical events:

```
# show events last 3 error
```

Show all events from 11:59:00 p.m. on Apr 30, 2008 through 11:59:00 a.m. on May 2, 2008:

```
# show events from 043008235900 to 050208115900
```

Show a range of events logged by controller A:

```
# show events from-event a100 to-event a123
```

See also

- `clear events`
- `set snmp-parameters`
- `show snmp-parameters`

show expander-status

Description **For use by or with direction from a service technician.**

Shows diagnostic information relating to SAS Expander Controller physical channels, known as PHY lanes. For each enclosure, this command shows status information for PHYs in I/O module A and then I/O module B.

Syntax `show expander-status`

Output `Encl`
Enclosure that contains the SAS expander.

`Ctlr`
I/O module that contains the SAS expander.

`Phy`
Identifies a PHY's logical location within a group based on the PHY type. Logical IDs are 0–23 for drive PHYs; 0–1 for SC PHYs; and 0–3 for other PHYs.

`Type`

- `Drive`: 1-lane PHY that communicates between the expander and a disk drive.
- `Egress`: 4-lane PHY that communicates between the expander and an expansion port or SAS Out port.
- `SC-1`: (Controller module only) 2-lane PHY that communicates between the expander and the partner's expander.
- `SC-0`: (Controller module only) 4-lane PHY that communicates between the expander and the SC.
- `Ingress`: (Expansion module only) 4-lane PHY that communicates between the expander and an expansion port.
- `Undefined`: No status information is available.
- `Unused`: Unused PHY.

`Status`

- `Enabled-Healthy`: The PHY is enabled and healthy.
- `Enabled-Degraded`: The PHY is enabled but degraded.
- `Disabled`: The PHY has been disabled by a user or by the system.

`Elem Status`
A standard SES status for the element:

- `Disabled`: Critical condition is detected.
- `Error`: Unrecoverable condition is detected. Appears only if there is a firmware problem related to PHY definition data.
- `Non-critical`: Non-critical condition is detected.
- `Not Used`: Element is not installed in enclosure.
- `OK`: Element is installed and no error conditions are known.
- `Unknown`: Either:
  - Sensor has failed or element status is not available. Appears only if an I/O module indicates it has fewer PHYs than the reporting I/O module, in which case all additional PHYs are reported as unknown.
  - Element is installed with no known errors, but the element has not been turned on or set into operation.

`Disabled`

- `Enabled`: PHY is enabled.
- `Disabled`: PHY is disabled.

`Reason`

- `0x8001`: PHY disabled because of error-count interrupts.
- `0x8003`: PHY disabled by PHY control page.
- `0x8004`: PHY is enabled but not ready. Appears for SC-1 PHYs when the partner I/O module is not installed. Appears for Drive, SC-1, or Ingress PHYs when a connection problem exists such as a broken connector.
- `0x8006`: PHY disabled because drive slot is empty.

Example Show expander status for a single-enclosure system with an empty disk slot:

```
# show expander-status
Encl   Ctlr Phy  Type         Status           Elem Status   Disabled  Reason
--------------------------------------------------------------------------------
0      A    0    Drive        Enabled-Healthy  OK            Enabled
0      A    1    Drive        Disabled         Disabled      Disabled
...
0      A    23   Drive        Disabled         OK            Disabled  0x8006
0      A    0    SC-1         Enabled-Healthy  OK            Enabled
0      A    1    SC-1         Enabled-Healthy  OK            Enabled
0      A    0    SC-0         Enabled-Healthy  OK            Enabled
...
0      A    3    SC-0         Enabled-Healthy  OK            Enabled
0      A    0    Egress       Enabled-Healthy  Non-critical  Enabled
...
0      A    3    Egress       Enabled-Healthy  Non-critical  Enabled
--------------------------------------------------------------------------------

Encl   Ctlr Phy  Type         Status           Elem Status   Disabled  Reason
--------------------------------------------------------------------------------
0      B    0    Drive        Enabled-Healthy  OK            Enabled
0      B    1    Drive        Enabled-Healthy  OK            Enabled
...
0      B    23   Drive        Disabled         OK            Disabled  0x8006
0      B    0    SC-1         Enabled-Healthy  OK            Enabled
0      B    1    SC-1         Enabled-Healthy  OK            Enabled
0      B    0    SC-0         Enabled-Healthy  OK            Enabled
...
0      B    0    Egress       Enabled-Healthy  Non-critical  Enabled
...
0      B    3    Egress       Enabled-Healthy  Non-critical  Enabled
--------------------------------------------------------------------------------
```

See also

- clear expander-status
- set expander-fault-isolation
- set expander-phy

show frus

Description Shows information for field-replaceable units (FRUs) in all enclosures. Some information reported is for use by service technicians.

Syntax `show frus`

Output `Name`

- `CHASSIS_MIDPLANE`: 2U chassis and midplane circuit board
- `RAID_IOM`: Controller module
- `BOD_IOM`: Expansion module
- `POWER_SUPPLY`: Power supply module

`Description`
FRU description

`Part Number`
FRU part number

`Serial Number`
FRU serial number

`Revision`
FRU revision number

`Dash Level`
FRU template revision number

`FRU Shortname`
FRU part number

`Manufacturing Date`
Date and time that the FRU was programmed

`Manufacturing Location`
Location where the FRU was programmed

`Manufacturing Vendor ID`
JEDEC ID of the manufacturer

`FRU Location`
Location of the FRU in the enclosure, as viewed from the back:

- `MID-PLANE SLOT`: Chassis midplane
- `UPPER IOM SLOT`: Controller module or expansion module A
- `LOWER IOM SLOT`: Controller module or expansion module B
- `LEFT PSU SLOT`: Power supply module 1
- `RIGHT PSU SLOT`: Power supply module 2

`Configuration SN`
Configuration serial number

`FRU Status`

- `Absent`: Component is not present
- `Fault`: One or more subcomponents has a fault
- `OK`: All subcomponents are operating normally
- `N/A`: Status is not available

show host-maps

Description Shows mapping information for volumes that are mapped to all hosts or to a specified host.

Syntax `show host-maps [host]`

Parameters *host*
Optional. Specifies the host's nickname or ID. If this parameter is omitted, mapping information for all hosts is shown

Output `ID`

- For FC and SAS: Host WWPN.
- For iSCSI: iSCSI host initiator's node name (typically the IQN).

`Name`
Host port nickname.

`Name`
Name of the volume seen by the host.

`Serial Number`
Serial number of the volume seen by the host.

`LUN`
LUN used to access the volume.

`Access`
Type of host access to the volume:

- `read-write`: Read and write
- `read-only`: Read only
- `none`: No access

`Ports`
Controller host ports that the mapping applies to.

Example Show volume mappings for all hosts:

```
# show host-maps
Host View [ID (AABBCCDDEEFF0011) Name (Host1) ] Mapping:
     Name   Serial Number                     LUN   Access       Ports
     ------------------------------------------------------------------
     vd2_v2 00c0ff28201500485b87a94803000000 1     read-write   A1,B0
     vd2_v0 00c0ff28201500485b87a94801000000 10    read-write   A0
     vd2_v1 00c0ff28201500485b87a94802000000 11    read-write   A0

  Host View [ID (1A2B3C4D5E6F8900) Name (Host2) ] Mapping:
     Name   Serial Number                     LUN   Access       Ports
     ------------------------------------------------------------------------
     vd2_v2 00c0ff28201500485b87a94803000000 0     read-write   A0,A1,B0,B1
     vd2_v0 00c0ff28201500485b87a94801000000 10    read-write   A0
     vd2_v1 00c0ff28201500485b87a94802000000 11    read-write   A0

  Host View [ID (210100E08B33340B) Name (Host3) ] Mapping:
     Name   Serial Number                     LUN   Access       Ports
     ------------------------------------------------------------------
     vd2_v2 00c0ff28201500485b87a94803000000 0     read-write   A0,B0
     vd2_v0 00c0ff28201500485b87a94801000000 10    read-write   A0
     vd2_v1 00c0ff28201500485b87a94802000000 11    read-write   A0
```

See also

- `show hosts`
- `show volume-maps`
- `show volumes`

show host-parameters

Description Shows information about host ports on both controllers.

Syntax `show host-parameters`

Output `Port`
Host port number

`Media`

- `FC(L)`: Fibre Channel-Arbitrated Loop
- `FC(P)`: Fibre Channel Point-to-Point
- `FC(-)`: Fibre Channel disconnected
- `SAS`: Serial Attached SCSI
- `iSCSI`: Internet SCSI

`Target ID`
Enclosure WWN or serial number

`Status`

- Up: Port is cabled and has an I/O link.
- Disconnected: Either no I/O link is detected or the port is not cabled.

`Speed (A)`
FC and SAS only. Actual link speed in Gbit/sec. Blank if not applicable.

`Speed (C)`
FC and SAS only. Configured link speed in Gbit/sec. Blank if not applicable.

`Topo (C)`
FC and SAS only. Configured topology.

`PID`
FC only. Primary ID. Blank if not applicable.

`Width`
SAS only. Number of SAS lanes configured for this port.

`IP Ver`
iSCSI only. IPv4

`PIP`
iSCSI only. Primary IP address

`PIP-Netmask`
iSCSI only. Primary subnet mask IP address

`PIP-Gateway`
iSCSI only. Primary gateway IP address

`PIP-Service Port`
iSCSI only. Service Port

`Port-Name`
iSCSI only. Target name for the port. The name is the storage system's IQN with the suffix `.a` for ports on controller A or `.b` for ports on controller B.

`Target Alias`
iSCSI only. Target alias for the port. The alias is the storage system's name with the suffix `.a` for ports on controller A or `.b` for ports on controller B.

Example Show host parameters for a controller with two FC ports:

```
# show host-parameters
Port Media    Target ID        Status    Speed(A) Speed(C) Topo(C) PID
------------------------------------------------------------------------
A0   FC(L)    207000C0FF000015 OK        2Gb      Auto      Loop    0

A1   FC(L)    217000C0FF000015 OK        2Gb      Auto      Loop    1
------------------------------------------------------------------------
```

Show host parameters for a controller with four SAS ports:

```
# show host-parameters
Port Media    Target ID        Status        Speed(A) Speed(C) Topo(C) Width
------------------------------------------------------------------------------
A0   SAS      500C0FF000013000 Disconnected                    Direct  4

A1   SAS      500C0FF000013100 Disconnected                    Direct  4

A2   SAS      500C0FF000013200 Disconnected                    Direct  4

A3   SAS      500C0FF000013300 Disconnected                    Direct  4
------------------------------------------------------------------------------
```

Show host parameters for a controller with two iSCSI ports:

```
# show host-parameters
  Port Media    Target ID        Status        Speed(A) Speed(C)
  ------------------------------------------------------------
  A0   iSCSI    IQN              Up            1Gb      1Gb
     IP-Ver PIP         PIP-Gateway PIP-Netmask MAC
     ------------------------------------------------------------
     IPv4   10.134.9.40 10.134.0.1  255.255.0.0 00:C0:FF:29:00:24

  A1   iSCSI    IQN              Disconnected          1Gb
     IP-Ver PIP         PIP-Gateway PIP-Netmask MAC
     ------------------------------------------------------------
     IPv4   10.134.9.41 10.134.0.1  255.255.0.0 00:C0:FF:29:00:23
-------------------------------------------------------------------------------
```

See also
- set host-parameters
- show ports

show hosts

Description Shows hosts that volumes can be mapped to. Host entries are automatically created for hosts that have sent an `inquiry` command or a `report luns` command to the system. Hosts typically do this when they boot up or scan for devices. When the command is received, the system saves the host port information; however, the information is retained after a restart only if you have set a nickname for the host.

Syntax `show hosts`

Output `Host ID`
Host WWPN or iSCSI node name

`Name`
Host nickname

`Discovered`

- `Yes`: The host was discovered and its entry was automatically created.
- `No`: The host entry was manually created.

`Mapped`
Shows whether the host is explicitly mapped to any volumes:

- `Yes`: At least one volume is explicitly mapped to the host.
- `No`: No volumes are explicitly mapped to the host.

Example Show hosts:

```
# show hosts
Host ID           Name     Discovered  Mapped
---------------------------------------------
100000A0B8040BAC  Host1    Yes         No
100000A0B8040BAD  My-host  Yes         Yes
1111111111111111  testhost No          No
---------------------------------------------
```

See also
- create host
- delete host
- set host-name

show iscsi-parameters

Description For iSCSI, shows system-wide iSCSI parameters.

Syntax `show iscsi-parameters`

Output `CHAP`
Whether Challenge-Handshake Authentication Protocol is enabled or disabled

`Jumbo Frames`
Whether jumbo-frame support is enabled or disabled

`iSNS`
Whether Internet Storage Name Service support is enabled or disabled

`iSNS IP`
Address of the iSNS server

`iSNS Alt IP`
Address of the alternate iSNS server

`iSCSI Speed`
Whether host port link speed is set to auto-negotiate (`Auto`) or to 1 Gbit/sec

Example Show system-wide iSCSI parameters:

```
# show iscsi-parameters
iSCSI Parameters
----------------
CHAP: Disabled
Jumbo Frames: Disabled
iSNS: Disabled
iSNS IP: 0.0.0.0
iSNS Alt IP: 0.0.0.0
iSCSI Speed: Auto
```

See also • `set iscsi-parameters`

show job-parameters

Description Shows parameters for background scrub, partner firmware upgrade, and other jobs.

Syntax `show job-parameters`

Output `Background Scrub`
Shows whether disks are automatically checked for disk defects to ensure system health.

`Partner Firmware Upgrade`
Shows whether component firmware versions are monitored and will be automatically upgraded on the partner controller.

`Utility Priority`
Priority at which jobs (such as vdisk verification and reconstruction but not background scrub) run with respect to I/O operations competing for the system's processors: `High`, `Medium`, or `Low`.

Example Show a system's job parameters:

```
# show job-parameters
Job Parameters
--------------
Background Scrub: Disabled
Partner Firmware Upgrade: Enabled
Utility Priority: High
```

See also • `set job-parameters`

show license

Description Shows the status of licensed features in the storage system.

Syntax `show license`

Output `License Key`

- The license key, if a license is installed and valid
- `not installed`, if a license is invalid or is not installed

`Base Maximum Snapshots`
Number of snapshots allowed without an installed license

`Licensed Snapshots`
Number of snapshots allowed by the installed license

`Maximum Licensable Snapshots`
Number of snapshots that the highest-level license allows

`In-Use Snapshots`
Number of existing snapshots

`Volume Copy`
Shows whether Volume Copy functions are enabled or disabled.

`VDS`
Shows whether the VDS (Virtual Disk Service) Hardware Provider is enabled or disabled.

`VSS`
Shows whether the VSS (Virtual Shadow Copy Service) Hardware Provider is enabled or disabled.

Example Show information about the installed license:

```
# show license
License Key: 0004b56810eb357d0f75d65c13c6e846
Base Maximum Snapshots: 16
Licensed Snapshots: 256
Maximum Licensable Snapshots: 256
In-Use Snapshots: 2
Volume Copy: Enabled
VDS: Disabled
VSS: Disabled
```

show master-volumes

Description Shows information about all master volumes, or ones associated with a specified controller or snap pool.

Syntax `show master-volumes [controller a|b] [snap-pool volume]`

Parameters `controller a|b`
Optional. Only includes master volumes owned by the specified controller.

`snap-pool volume`
Optional. Only includes master volumes associated with the specified snap pool name or serial number. For syntax, see Command syntax.

Output `Vdisk`
Vdisk name

`Serial Number`
Master volume serial number

`Name`
Master volume name

`Size`
Total size of the master volume

`Status`
Indicates whether the master volume is available or unavailable

`Status-Reason`
Shows `---` for `Available` status, or a reason for `Unavailable` status:

- `MV Not Accessible`: Master volume is not accessible
- `SP Not Accessible`: Snap pool is not accessible
- `SP Not Found`: Snap pool is not found
- `Unknown`

`Snap-pool Name`
Name of the associated snap pool

`Snapshots`
Number of snapshots that exist for the master volume

`Snap Data`
Amount of snap-pool space occupied by this master volume for its associated snapshots (preserved and write data)

`Rollback`
Either the percent complete if rollback is in progress, or `---` if rollback is not in progress

Example Show information about master volumes associated with snap pool `SP1`:

```
# show master-volumes snap-pool SP1
Vdisk  Serial Number  Name   Size    Status     Status-Reason  Snap-pool Name
  Snapshots  Snap Data  Rollback
------------------------------------------------------------------------------
VD1    SN             MV1    20.0GB  Available  --             SP1
  1          0B         ---
------------------------------------------------------------------------------
```

See also

- convert master-to-std
- create master-volume
- delete all-master-volumes
- delete master-volume
- expand volume
- rollback master-volume

show network-parameters

Description Shows the network settings for each RAID controller.

Syntax

```
show network-parameters
```

Output `IP Address`
Controller IP address

`Gateway`
Controller gateway

`Subnet Mask`
Controller subnet mask

`MAC Address`
Controller's unique Media Access Control address

`Addressing Mode`

- `Manual`: Network settings set manually (statically)
- `DHCP`: DHCP used to set network parameters

Example Show network parameters for a storage system using DHCP:

```
# show network-parameters
Network Parameters Controller A
-------------------------------
IP Address: 10.134.129.188
Gateway: 10.134.0.1
Subnet Mask: 255.255.0.0
MAC Address: 00:C0:FF:0A:A3:26
Addressing Mode: DHCP

Network Parameters Controller B
-------------------------------
IP Address: 10.134.129.189
Gateway: 10.134.0.1
Subnet Mask: 255.255.0.0
MAC Address: 00:C0:FF:0A:A3:14
Addressing Mode: DHCP
```

Show network parameters for a storage system using manual addressing:

```
# show network-parameters
Network Parameters Controller A
-------------------------------
IP Address: 172.22.1.200
Gateway: 172.22.1.1
Subnet Mask: 255.255.255.0
MAC Address: 00:C0:FF:0A:43:18
Addressing Mode: Manual

Network Parameters Controller B
-------------------------------
IP Address: 172.22.1.201
Gateway: 172.22.1.1
Subnet Mask: 255.255.255.0
MAC Address: 00:C0:FF:0A:43:26
Addressing Mode: Manual
```

See also

- `set network-parameters`

show ntp-status

Description Shows the status of the use of Network Time Protocol (NTP) in the system.

Syntax `show ntp-status`

Output `Status`

- `activated`: NTP is enabled
- `deactivated`: NTP is disabled

`Client Task Status`

- `n/a`: NTP is disabled
- `present`: NTP is enabled and the client task is active
- `missing`: NTP is enabled but the client task is in an interim state

`NTP Server Address`
NTP server IP address, if set

`Last Server Contact`
Date and time, in UT, of the last message received from the NTP server, if any

Example Show NTP status for the system:

```
# show ntp-status
NTP Status
----------
Status: activated
Client Task Status: present
NTP Server Address: 69.10.36.3
Last Server Contact: 2008-12-04 16:24:42
```

See also
- `set controller-date`

show ports

Description Shows information about host ports on both controllers.

Syntax `show ports`

Output `Port`
Controller ID and port number

`Media`

- `FC(P)`: Fibre Channel Point-to-Point
- `FC(L)`: Fibre Channel-Arbitrated Loop
- `FC(-)`: Not applicable, as when the port is disconnected
- `SAS`: Serial Attached SCSI
- `iSCSI`: Internet SCSI

`Target ID`
Enclosure WWN or serial number

`Status`
Whether the port is operational, has a problem, or is disconnected

`Speed (A)`
Actual host-port link speed, or blank if not applicable.

`Speed (C)`
FC and SAS only. Configured host-port link speed:

- FC: `2Gb` or `4Gb` (Gbit/sec)
- SAS: `1Gb` (Gbit/sec)
- Blank if not applicable

`Topo (C)`
FC and SAS only. Configured topology

`Width`
SAS only. Number of SAS lanes in this port.

`PID`
FC and SAS only. Primary ID, or blank if not applicable

`IP-Ver`
iSCSI only. IPv4

`PIP`
iSCSI only. Primary IP address

SIP
iSCSI only. Secondary IP address. This field is only displayed during failover.

MAC
iSCSI only. Unique Media Access Control (MAC) hardware address, also called the physical address.

Example Show port information for a system with two FC ports:

```
# show ports
Port Media    Target ID        Status      Speed(A) Speed(C) Topo(C) PID
-------------------------------------------------------------------------
A0   FC(L)    207000C0FF000015 OK          2Gb      2Gb      Loop    0

A1   FC(L)    217000C0FF000015 OK          2Gb      2Gb      Loop    1
-------------------------------------------------------------------------
```

Show port information for a system with four SAS ports:

```
# show ports
Port Media    Target ID        Status       Speed(A) Speed(C) Topo(C) Width
-------------------------------------------------------------------------
A0   SAS      500C0FFD41011000 Disconnected                   Direct  4

A1   SAS      500C0FFD41011100 Disconnected                   Direct  4

A2   SAS      500C0FFD41011200 Disconnected                   Direct  4

A3   SAS      500C0FFD41011300 Disconnected                   Direct  4
-------------------------------------------------------------------------
```

See also • set host-parameters

show protocols

Description Shows which management services and protocols are enabled or disabled.

Syntax `show protocols`

Example Show the status of service and security protocols:

```
# show protocols
Service and Security Protocols
------------------------------
Web Browser Interface (HTTP): Enabled
Secure Web Browser Interface (HTTPS): Enabled
Command Line Interface (Telnet): Enabled
Secure Command Line Interface (SSH): Enabled
Storage Management Initiative Specification (SMIS): Enabled
File Transfer Protocol (FTP): Disabled
Simple Network Management Protocol (SNMP): Enabled
Service Interface (Service): Disabled
Service Debug (Debug): Disabled
In-band SES Management (SES): Enabled
In-band CAPI Management (CAPI): Enabled
```

See also • set protocols

show redundancy-mode

Description Shows the redundancy status of the system.

Syntax `show redundancy-mode`

Output `Redundancy Mode`
The redundancy mode, also called the operation mode.

- `Active-Active ULP`: Both controllers are active using ULP (Unified LUN Provisioning).
- `Single-Controller`: The enclosure contains a single controller using ULP.

`Redundancy Status`

- `Redundant`: Both controllers are operational.
- `Operational but not redundant`: In active-active mode, one controller is operational and the other is offline. In single-controller mode, the controller is operational.

`Controller` *`ID`* `Status`

- `Operational`: The controller is operational.
- `Down`: The controller is installed but not operational.
- `Not Installed`: The controller is not installed.

`Controller` *`ID`* `Serial Number`

- Controller module serial number
- `Not Available`: The controller is down or not installed.

Example Show the redundancy status of a dual-controller system where controller A is offline:

```
# show redundancy-mode
System Redundancy
-----------------
Redundancy Mode: Active-Active ULP
Redundancy Status: Operational but not redundant
Controller A Status: Down
Controller A Serial Number: Not Available
Controller B Status: Operational
Controller B Serial Number: 00C0FF0A4326
```

Show the redundancy status of an operational dual-controller system using ULP:

```
# show redundancy-mode
System Redundancy
-----------------
Redundancy Mode: Active-Active ULP
Redundancy Status: Redundant
Controller A Status: Operational
Controller A Serial Number: 00C0FF0A4318
Controller B Status: Operational
Controller B Serial Number: 00C0FF0A4326
```

show refresh-counters

Description For API use, shows when the data represented by the base type was last updated.

Syntax `show refresh-counters`

show sas-link-health

Description Shows the condition of SAS port connections.

Syntax `show sas-link-health`

Output `Encl`
Enclosure ID

`Ctlr`
ID of the controller module or expansion module

`Name`

- `Out Port`: Egress (expansion) port on controller module or an expansion module. Can be connected to an ingress port on an expansion module.
- `In Port`: Ingress port on an expansion module. Can be connected to an egress (expansion) port on a controller module or an expansion module.

`Status`
Status of the connection:

- `OK`
- `Disconnected`
- `Not Present`
- `Warning`
- `Error`
- `Unknown`

`Health`
Relative status of the connection:

- `OK`
- `Degraded`
- `Fault`
- `Unknown`

Example Show the condition of SAS port connections in a two-enclosure system:

```
# show sas-link-health
Encl     Ctlr     Name                 Status               Health
------------------------------------------------------------------------
0        A        Out Port             OK                   OK
0        B        Out Port             OK                   OK
------------------------------------------------------------------------

Encl     Ctlr     Name                 Status               Health
------------------------------------------------------------------------
1        A        In Port              OK                   OK
1        A        Out Port             Disconnected         N/A
1        B        In Port              OK                   OK
1        B        Out Port             Disconnected         N/A
------------------------------------------------------------------------
```

show schedule-details

Description Shows details for a specified task schedule.

Syntax `show schedule-details` *schedule*

Parameters *schedule*
Specifies the schedule name

Example Show details for task schedule `Sched1`, which should next run task `T1` on February 23rd at 12:59:

```
# show schedule-details Sched1
Schedule Details
----------------
Schedule Name: Sched1
Schedule Specification: Start 2/19/2007 23:47:00, Every 3 Minutes
Schedule Status: Ready
Next Time: 2/23/2007 12:59:00
Task To Run: T1
Error Message: none

 Tasks
 -----
 Task Name: T1
 Task Type: TakeSnapshot
 Task Status: Ready
 Task State: Init

   Source Volume Source Volume Serial             Prefix  Count Last Created
   ------------------------------------------------------------------------------
   VD1_V1        00c0ff2824460048904eba4901000000 T1      1     none
      Snapshot Name        Snapshot Serial
      ------------------------------------------------------
      T1_S0001             00c0ffd2710700482ce3de4501000000
```

See also
- create schedule
- delete schedule
- show schedules

show schedules

Description Shows details for all task schedules.

Syntax `show schedules`

Example Show details for the two existing task schedules:

```
# show schedules
Schedules
---------
Schedule Name: Sched1
Schedule Specification: Start 2/19/2007 23:47:00, Every 3 Minutes
Schedule Status: Ready
Next Time: 2/23/2007 12:59:00
Task To Run: T1
Error Message: none

 Tasks
 -----
 Task Name: T1
 Task Type: TakeSnapshot
 Task Status: Ready
 Task State: Init

   Source Volume Source Volume Serial             Prefix  Count Last Created
   ------------------------------------------------------------------------------
   VD1_V1        00c0ff2824460048904eba4901000000 T1      1     T1_S0001
      Snapshot Name        Snapshot Serial
      ------------------------------------------------
      T1_S0001             00c0ff28244600487624d64901000000

Schedule Name: v1_v001_sched1
...
```

See also
- create schedule
- delete schedule
- show schedule-details

show sensor-status

Description Shows temperatures and voltages of controller modules and power supplies. These values are reported by environmental sensors in each controller module (Ctlr, IOM) and power supply (PSU).

Syntax `show sensor-status`

Output `Sensor Name`
Sensor name and location

`Value`
The value of the sensor

`Status`

- `Absent`: Component is not present
- `Fault`: One or more subcomponents has a fault
- `OK`: All subcomponents are operating normally. Temperature status OK indicates that the sensor is working properly, not that the temperature is within an acceptable range.
- `N/A`: Status is not available

Example Show temperature and voltage status for the controllers and power supplies:

```
# show sensor-status
Sensor Name                          Value    Status
-------------------------------------------------
CPU Temperature-Ctlr A               41       OK
CPU Temperature-Ctlr B               58       OK
FPGA Temperature-Ctlr A              48       OK
FPGA Temperature-Ctlr B              52       OK
Onboard Temperature 1-Ctlr A         34       OK
Onboard Temperature 1-Ctlr B         36       OK
Onboard Temperature 2-Ctlr A         40       OK
Onboard Temperature 2-Ctlr B         47       OK
Capacitor Temperature-Ctlr A         29       OK
Capacitor Temperature-Ctlr B         29       OK
Capacitor Pack Voltage-Ctlr A        8.16     OK
Capacitor Pack Voltage-Ctlr B        8.14     OK
Capacitor Cell 1 Voltage-Ctlr A      2.02     OK
Capacitor Cell 1 Voltage-Ctlr B      2.02     OK
Capacitor Cell 2 Voltage-Ctlr A      2.02     OK
Capacitor Cell 2 Voltage-Ctlr B      2.02     OK
Capacitor Cell 3 Voltage-Ctlr A      2.06     OK
Capacitor Cell 3 Voltage-Ctlr B      2.03     OK
Capacitor Cell 4 Voltage-Ctlr A      2.06     OK
Capacitor Cell 4 Voltage-Ctlr B      2.08     OK
Capacitor Charge-Ctlr A              100%     OK
Capacitor Charge-Ctlr B              100%     OK
Overall Unit Status                  Warning  Warning
Temperature Loc:upper-IOM A          38       OK
Temperature Loc:lower-IOM B          40       OK
Temperature Loc:left-PSU 1           28       OK
Temperature Loc:right-PSU 2          34       OK
Voltage 12V Loc:upper-IOM            11.92    OK
Voltage 5V Loc:upper-IOM             5.08     OK
Voltage 12V Loc:lower-IOM            11.81    OK
Voltage 5V Loc:lower-IOM             5.08     OK
Voltage 12V Loc:left-PSU 1           12.08    OK
Voltage 5V Loc:left-PSU 1            5.13     OK
Voltage 3.3V Loc:left-PSU 1          3.49     OK
Voltage 12V Loc:right-PSU 2          12.02    OK
Voltage 5V Loc:right-PSU 2           5.16     OK
Voltage 3.3V Loc:right-PSU 2         3.49     OK
Current 12V Loc:upper-IOM            3.89     OK
Current 12V Loc:lower-IOM            4.95     OK
Current 12V Loc:left-PSU 1           6.15     OK
Current 5V Loc:left-PSU 1            6.67     OK
Current 12V Loc:right-PSU 2          6.15     OK
Current 5V Loc:right-PSU 2           7.02     OK
-------------------------------------------------
```

See also • show enclosure-status

show shutdown-status

Description Shows whether each Storage Controller is active or shut down.

Syntax `show shutdown-status`

Example Show the shutdown status for each controller:

```
# show shutdown-status
storage controller A is up
storage controller B is down
```

See also
- restart
- shutdown

show snap-pools

Description Shows information about snap pools owned by a specified controller or both controllers.

Syntax `show snap-pools [controller a|b|both]`

Parameters

`controller a|b|both`
Optional. Shows snap pools owned by controller A only, by controller B only, or by either controller (`both`). If this parameter is omitted, all snap pools are shown.

Output

`Vdisk`
Vdisk name

`Serial#`
Snap pool serial number

`Name`
Snap pool name

`Size`
Total size of the snap pool volume

`Free`
Amount of free space available in this snap pool

`MasterVols`
Number of master volumes associated with this snap pool

`Snapshots`
Number of snapshots using this snap pool

`Threshold`
Snap pool threshold level (Warning, Error, and Critical)

`%Usage`
Threshold value (percent of snap pool space used) that triggers the threshold's policy

`Policy`
Recovery policy invoked when threshold value is reached

`SizeToExpand`
Increment size by which the snap pool is automatically expanded each time the threshold level is reached. This parameter applies when the `AutoExpand` policy is active; otherwise its value is `N/A`.

Example Show information for snap pools owned by controller A:

```
# show snap-pools controller a
Vdisk  Serial Number  Name  Size      Free      MasterVols  Snapshots
--------------------------------------------------------------------
R5     SN             SP1   374.6GB   374.6GB   1           2
     Threshold  %Usage   Policy                 SizeToExpand
     ---------------------------------------------------------
     Warning    75%      Notify Only            N/A
     Error      90%      Auto Expand            5013.5KB
     Critical   99%      Delete Snapshots       N/A
--------------------------------------------------------------------
```

See also

- create snap-pool
- delete snap-pool
- expand snap-pool
- set snap-pool-policy
- set snap-pool-threshold

show snapshots

Description Shows information about snapshots for a specified controller, master volume, or snap pool. If no parameters are specified, information about all snapshots is shown.

Syntax To show snapshots by controller:

```
show snapshots [controller a|b|both]
```

To show snapshots by master volume:

```
show snapshots [master-volume volume]
```

To show snapshots by snap pool:

```
show snapshots [snap-pool volume]
```

Parameters

`controller a|b|both`
Optional. Shows snapshots owned by controller A only, by controller B only, or by either controller (`both`).

`master-volume` *`volume`*
Optional. Shows snapshots associated with the specified master volume name or serial number. For syntax, see Command syntax.

`snap-pool` *`volume`*
Optional. Shows snapshots associated with the specified snap pool name or serial number. For syntax, see Command syntax.

Output

`Vdisk`
Vdisk name

`Serial Number`
Snapshot serial number

`Name`
Snapshot name

`Creation Date/Time`
Date and time the snapshot was prepared or committed

`Status`
Indicates whether the snapshot is available or unavailable

`Status-Reason`
Shows `---` for `Available` status, or a reason for `Unavailable` status:

- `MV Not Accessible`: Master volume is not accessible
- `MV Not Found`: Master volume is not found
- `SP Not Accessible`: Snap pool is not accessible
- `SP Not Found`: Snap pool is not found
- `SS Pending`: Snapshot is pending
- `VC-MD In Progress`: Volume-copy with modified data is in progress
- `RB-MD In Progress`: Rollback with modified data is in progress
- `Unknown`

`Master Volume Name`
Name of the volume that the snapshot was taken of

`Snap-pool Name`
Snap pool name

`Snap Data`
Total amount of preserved and write data associated with the snapshot

`UniqueData`
Amount of preserved and write data that is unique to the snapshot

`SharedData`
Amount of preserved and write data that is shared between this snapshot and other snapshots

Example Show information about snapshots associated with snap pool `SP2`:

```
# show snapshots snap-pool SP2
Vdisk  Serial Number  Name  Creation Date/Time   Status     Status-Reason
  Master Volume Name   Snap-pool Name  Snap Data  UniqueData  SharedData
------------------------------------------------------------------------------
VD1    SN             SS1   2008-03-19 13:32:11  Available  ---
  V2                   SP2             0B         0B          0B
------------------------------------------------------------------------------
```

See also
- `show master-volumes`
- `show snap-pools`

show snmp-parameters

Description Shows SNMP settings for event notification.

Syntax `show snmp-parameters`

Output `SNMP`

- `enabled`: SNMP notification is enabled
- `disabled`: SNMP notification is disabled

To enable or disable requests to the SNMP agent use the `set protocols` command.

`SNMP Filter`

- `crit`: Only critical events are sent as traps
- `warn`: All critical events and warnings are sent as traps
- `info`: All events are sent as traps
- `none`: No events are sent as traps and traps are disabled

`SNMP Trap Host IP#`
IP address of each trap host

`SNMP read community`
Community string for read-only access (not shown to Monitor users)

`SNMP write community`
Community string for write access (not shown to Monitor users)

Example Show SNMP notification settings:

```
# show snmp-parameters
SNMP: enabled
SNMP Filter: crit
SNMP Trap Host IP1: 172.22.4.171
SNMP Trap Host IP2: 0.0.0.0
SNMP Trap Host IP3: 0.0.0.0
SNMP read community: public
SNMP write community: private
```

See also
- `set snmp-parameters`
- `show protocols`

show system

Description Shows information about the system.

Syntax `show system`

Example Show system information, including installed languages (supported locales):

```
# show system
System Information
------------------
System Name: Storage-1
System Contact: J. Smith
System Location: Main lab
System Information: Used for order processing data
Vendor Name: vendor
Product ID: product
Product Brand: brand
SCSI Vendor ID: vendor-ID
Enclosure Count: 2
Health: OK
Supported Locales: English
```

See also
- `set system`

show system-parameters

Description Shows the current settings for the storage system.

Syntax `show system-parameters`

Output `ULP`
Shows that Unified LUN Provisioning is enabled

`Number of Host Ports`
Number of host-interface ports in the controller enclosure

`Maximum Disks`
Number of disks that the system supports

`Maximum Volumes`
Number of volumes that the system supports

`Maximum Vdisks`
Number of vdisks that the system supports

`Maximum LUNs`
Number of LUNs that the system supports

`Maximum Vdisks Per Controller`
Number of vdisks that each controller supports

`Local Controller`
ID of the controller you are accessing

`Serial Number`
Last five digits of the midplane serial number.

Example Show current settings for the storage system:

```
# show system-parameters
System Parameters
-----------------
ULP Enabled: true
Number of Host Ports: 4
Maximum Disks: 108
Maximum Volumes: 512
Maximum Vdisks: 32
Maximum LUNs: 512
Maximum Vdisks Per Controller: 16
Local Controller: A
Serial Number: 000013
```

See also
- `set volume`
- `set vdisk`

show task-details

Description Shows details for all tasks or a specified task.

Syntax `show task-details [task]`

Parameters *task*
Optional. Specifies a task name.

Output **For a TakeSnapshot task:**

`Task Name`
Task name

`Task Type`
`TakeSnapshot`

`Task Status`

- `Ready`: Task is ready to run
- `Active`: Task is currently running

`Task State`
Current state of the task: `Init`, `Vol Verified`, `License Checked`, `Name Created`, `Snap Created`, `Snap Verified`

`Source Volume`
Standard or master volume name

`Source Volume Serial`
Volume serial number

`Prefix`
Label that identifies snapshots created by this task. Snapshot names have the format *`prefix`*`_s001` through *`prefix`*`_s1023`.

`Count`
Number of snapshots to retain with this prefix. When a new snapshot exceeds this limit, the oldest snapshot with the same prefix is deleted.

`Last Created`
Name of the last snapshot taken, or none

`Snapshot Name`
Name of each snapshot taken, or blank

`Snapshot Serial`
Serial number of each snapshot taken, or blank

`Error Message`
Any error message associated with this task

For a ResetSnapshot task:

`Task Name`
Task name

`Task Type`
`ResetSnapshot`

`Task Status`

- `Ready`: Task is ready to run
- `Active`: Task is currently running

`Task State`
Current state of the task: `Init`, `Snap Verified`

`Snapshot Name`
Name of the snapshot to reset

`Snapshot Serial`
Serial number of the snapshot to reset

`Error Message`
Any error message associated with this task

For a VolumeCopy task:

`Task Name`
Task name

`Task Type`
`VolumeCopy`

`Task Status`

- `Ready`: Task is ready to run
- `Active`: Task is currently running

`Task State`
Current state of the task: `Init`, `Vol Verified`, `Name Created`, `Vol Created`

`Source Volume Name`
Name of the volume to be copied

`Source Volume Serial`
Serial number of the volume to be copied

`Destination Vdisk Name`
Name of the destination vdisk

`Destination Vdisk Serial`
Serial number of the destination vdisk

`Destination Volume Prefix`
Label that identifies copies created by this task. Volume names have the format *prefix*`_c001` through *prefix*`_c1023`.

`Include Modified Data`
True or False

`Last Copy Created`
Name of the last copy created

`Error Message`
Any error message associated with this task

Example Show details for task `Snap`:

```
# show task-details Snap
Tasks
-----
Task Name: Snap
Task Type: TakeSnapshot
Task Status: Ready
Task State: Init

  Source Volume  Source Volume Serial  Prefix  Count  Last Created
  ------------------------------------------------------------------
  VD1            SN                    VD1     1      VD1_S0001
     Snapshot Name  Snapshot Serial
     ------------------------------
    VD1_S0001    SN
```

See also

- `create schedule`
- `create task`
- `delete task`
- `show schedules`
- `show tasks`

show tasks

Description Shows details for all tasks.

Syntax `show tasks`

Example Show details for the two existing tasks:

```
# show tasks
Tasks
-----
Task Name: Snap
Task Type: TakeSnapshot
Task Status: Ready
Task State: Init

  Source Volume  Source Volume Serial  Prefix  Count  Last Created
  ------------------------------------------------------------------
  VD1            SN                   VD1     1      VD1_S0001
     Snapshot Name  Snapshot Serial
     ------------------------------
     VD1_S0001      SN

Task Name: v1_v002_task001
...
```

See also

- create task
- create schedule
- delete task
- show schedule-details
- show task-details

show users

Description Shows configured user profiles.

Syntax `show users [user]`

Input `user`
Optional. User name to show settings for.

Output `Username`
User name

`Access Level`

- `Monitor`: View-only access to selected user interfaces
- `Manage`: Modify access to selected user interfaces

`User Type`
Applies to RAIDar only.

- `Standard`: Has access to standard administrative functions
- `Advanced`: Has access to standard and advanced functions
- `Diagnostic`: Has access to standard, advanced, and troubleshooting functions

`User Locale`
Display language for this user

`WBI`
`x` indicates user can use the web-browser interface

`CLI`
x indicates user can use the command-line interface

`FTP`
x indicates user can use the file transfer protocol interface

Example Show configured users:

```
# show users
Username   Access Level  User Type    User Locale  WBI   CLI   FTP
-------------------------------------------------------------------
manage     Manage        Standard     English      x     x     x
monitor    Monitor       Advanced     English      x     x
ftp        Manage        Diagnostic   English                  x
-------------------------------------------------------------------
```

See also
- `create user`
- `delete user`
- `set user`

show vdisks

Description Shows information for all or specified vdisks.

Syntax `show vdisks [vdisks]`

Parameters *vdisks*
Optional. Names or serial numbers of the vdisks to show information about. For syntax, see Command syntax.

Output `Name`
Vdisk name

`Size`
Vdisk size

`Free`
Vdisk free space

`Own`
Either the preferred owner during normal operation or the partner controller when the preferred owner is offline

`Pref`
Controller that owns the vdisk and its volumes during normal operation

`RAID`
Vdisk RAID level

`Disks`
Number of disks in the vdisk

`Spr`
Number of spares assigned to the vdisk

`Chk`

- For RAID levels except RAID 50, the configured chunk size for the vdisk.
- For RAID 50, the vdisk chunk size calculated as: *configured-chunk-size* x (*subvdisk-members* - 1). For a vdisk configured to use 32-KB chunk size and 4-disk subvdisks, the value would be `96k` (32KB x 3).

Status

- CRIT: The vdisk is online, however some disks are down and the vdisk is not fault tolerant
- FTDN: The vdisk is online and fault tolerant, however some of the disks are down
- FTOL: The vdisk is online and fault tolerant
- OFFL: The vdisk is offline because it is using offline initialization, or because disks are down and data may be lost
- QTCR: The vdisk is in a critical state and has been quarantined because some disks are missing
- QTOF: The vdisk is offline and has been quarantined because some disks are missing
- UP: The vdisk is online and does not have fault-tolerant attributes

Jobs
Shows whether a job is running and its percent complete.

- EXPD: Vdisk is being expanded
- INIT: Vdisk is initializing
- RCON: Vdisk is being reconstructed
- VRFY: Vdisk is being verified
- VRSC: Vdisk is being scrubbed

Serial Number
Vdisk serial number

Example Show information about vdisk VD1 only:

```
# show vdisks VD1
Name  Size    Free    Own Pref  RAID  Disks  Spr  Chk  Status Jobs
  Serial Number
------------------------------------------------------------------------
VD1   587.1GB 116.7GB B   B     RAID50 6     0    128k FTOL   VRSC 41%
  SN
------------------------------------------------------------------------
```

See also

- create vdisk
- delete vdisk
- expand vdisk
- set vdisk

show volume-maps

Description Shows mapping information for a specified volume or for all volumes.

Syntax `show volume-maps [volume]`

Parameters *volume*
Optional. Name or serial number of the volume to show mappings for. For syntax, see Command syntax. If this parameter is omitted, information for all volumes is shown.

Output Serial Number
Volume serial number

Name
Volume name

Ports
Controller host ports that the mapping applies to

LUN
LUN used to access the volume

`Access`
Type of host access to the volume:

- `read-write`
- `read-only`
- `no-access`
- `not-mapped`

`Host-Port-Identifier`

- FC and SAS: Host WWPN
- iSCSI: Host initiator node name (typically the IQN)
- `all other hosts` for the volume's default mapping

`Nickname`
Host nickname, or blank if not set or for `all other hosts`

Example Show all volume mappings:

```
# show volume-maps
Info: Retrieving data...
Volume View [Serial Number (SN) Name (v1) ] Mapping:
  Ports       LUN   Access       Host-Port-Identifier Nickname
  ------------------------------------------------------------
  A0,A1,B0,B1 501   read-write   all other hosts

Volume View [Serial Number (SN) Name (v2) ] Mapping:
  Ports LUN   Access       Host-Port-Identifier Nickname
  ------------------------------------------------------
              not-mapped   all other hosts
```

See also

- `show host-maps`
- `show hosts`
- `show volumes`

show volumecopy-status

Description Shows information about in-progress volume copy operations. While a volume copy is in progress, the destination volume cannot be accessed.

Syntax

```
show volumecopy-status [controller a|b]
```

Parameters `controller a|b`
Optional. Shows volume copy operations for volumes owned by controller A or B only. If this parameter is omitted, all volume copy operations are shown.

Output `VC Volume Name`
Destination volume name

`Serial#`
Destination volume serial number

`Vdisk`
Destination vdisk name

`Source Volume`
Source volume name

`Progress`
Percent complete of the volume copy

`Status`
Indicates whether the destination volume is Unavailable or Suspended

`Status-Reason`
The status is `Unavailable` while the volume-copy is in progress. The status is `Suspended` if the source volume goes offline while the copy is in progress. When the source volume comes back online, the copy process resumes from the point where it stopped.

Example Show information about volume copies in progress for controller A:

```
# show volumecopy-status controller a
VC Volume Name  Serial Number  Vdisk  Source Volume  Progress  Status
  Status-Reason
------------------------------------------------------------------------------
MV1-copy        SN             VD1    MV1            7%        Unavailable
  VC In Progress
------------------------------------------------------------------------------
```

See also
- `abort volumecopy`
- `volumecopy`

show volumes

Description Shows volume information for all or specified vdisks.

Syntax
```
show volumes
    [vdisk vdisks]
    [class standard|ptsnap]
    [type snap-pool|mastervolume|snapshot|standard]
```

Parameters

`vdisk vdisks`
Optional. Names or serial numbers of the vdisks containing the volumes to show. For syntax, see Command syntax.

`class standard|ptsnap`
Optional. Specifies the class of volumes to show.

`type snap-pool|mastervolume|snapshot|standard`
Optional. Specifies the type of volumes to show.

Output

`Vdisk`
Name of the vdisk

`Name`
Name of the volume

`Size`
Volume size

`Serial Number`
Volume serial number

`WR Policy`
Write-back cache mode (write-back or write-through)

`Cache Opt`
Read-ahead cache mode (standard or super-sequential)

`Read Ahead Size`

- `Disabled`
- `Default`
- `Maximum`
- `Specific size`

Type

- standard: Standard volume
- standard*: Destination of an in-progress volume copy and cannot be mounted until the copy is complete
- snap-pool: Snap-pool volume
- mastervol: Master volume
- snapshot: Snapshot volume
- unknown: Unknown

Class
Standard, PTSNAP (snapshot-related), or unknown

Example Show volume information for standard volumes only:

```
# show volumes type standard
Vdisk  Name  Size    Serial Number  WR Policy   Cache Opt  Read Ahead Size
  Type       Class
------------------------------------------------------------------------------
VD1    V0    20.0GB  SN             write-back  standard   Default
  standard   standard
------------------------------------------------------------------------------
```

Show volume information for vdisk VD1 only:

```
# show volumes vdisk VD1
Vdisk  Name  Size    Serial Number  WR Policy   Cache Opt  Read Ahead Size
  Type       Class
------------------------------------------------------------------------------
VD1    V0    35.9GB  SN             write-back  standard   Default
  standard   standard
VD1    V1    35.9GB  SN             write-back  standard   Default
  mastervol  PTSNAP
VD1    V2    35.9GB  SN             write-back  standard   Default
  snap-pool  PTSNAP
VD1    V3    35.9GB  SN             write-back  standard   Default
  snapshot   PTSNAP
------------------------------------------------------------------------------
```

See also

- create volume
- delete volume
- expand volume
- set volume
- show vdisks
- show volume-maps

shutdown

Description Shuts down the Storage Controller in a controller module. This ensures that a proper failover sequence is used, which includes stopping all I/O operations and writing any data in write cache to disk. If the Storage Controller in each controller module is shut down, hosts cannot access the system's data. Perform a shut down before removing a controller module or powering down the system.

△ **CAUTION:** You can continue to use the CLI when either or both Storage Controllers are shut down, but information shown might be invalid.

Syntax

```
shutdown a|b|both
```

Parameters

`a|b|both`
Specifies to shut down the Storage Controller in controller A, B, or both.

Example Shut down the Storage Controller in controller A:

```
# shutdown a
Info: Shutting down SC a...
Success: Command completed successfully
```

See also

- restart
- show shutdown-status

stty

Description Sets and shows terminal information.

Syntax

```
stty info | hardwrap | rows # | columns #
```

Parameters

`info`
Shows the terminal's settings.

`hardwrap`
Enables or disables the hard wrapping of output. Terminals usually wrap at the screen width without truncating output, but turning on hard wrapping ensures this.

`rows #`
Sets the number of rows that a terminal can display. The terminal usually sets this value; this is an override. The info parameter shows this as screen height.

`columns #`
Sets the number of columns that a terminal can display. The terminal usually sets this value; this is an override. The info parameter shows this as screen width.

Example Show information about the terminal:

```
# stty info
Terminal Type: ANSI
Screen width : 140
Screen height: 60
Hard wrap    : Off
Success: Command completed successfully.
```

test

Description Sends a message to test event notification. After issuing this command, verify that test message reached the configured destinations.

Syntax `test email|snmp|notification`

Parameters `email`
Sends a test message to configured email addresses.

`snmp`
Sends a test message to configured SNMP trap hosts.

`notification`
Sends a test message to configured email addresses and SNMP trap hosts.

Example Test email and SNMP notification of events:

```
# test notification
Success: Command completed successfully.
```

See also
- `set email-parameters`
- `set snmp-parameters`

trust

Description Enables an offline vdisk to be brought online for emergency data collection. This command must be enabled before each use.

△ **CAUTION:** This command can cause unstable operation and data loss if used improperly. It is intended for disaster recovery only.

The trust command resynchronizes the time and date stamp and any other metadata on a bad disk disk. This makes the disk an active member of the vdisk again. You might need to do this when:

- One or more disks in a vdisk start up more slowly or were powered on after the rest of the disks in the vdisk. This causes the date and time stamps to differ, which the system interprets as a problem with the "late" disks. In this case, the vdisk functions normally after being trusted.
- A vdisk is offline because a disk is failing, you have no data backup, and you want to try to recover the data from the vdisk. In this case, trust may work, but only as long as the failing disk continues to operate.

When the "trusted" vdisk is back online, back up its data and audit the data to make sure that it is intact. Then delete that vdisk, create a new vdisk, and restore data from the backup to the new vdisk. Using a trusted vdisk is only a disaster-recovery measure; the vdisk has no tolerance for any additional failures.

Syntax To enable the `trust` command:

`trust enable`

To trust a vdisk:

`trust vdisk` *`vdisk`*

Parameters `enable`
Enables the trust command before use.

`vdisk` *`vdisk`*
Name or serial number of the vdisk to trust. For syntax, see Command syntax.

Example Enable the trust command and then trust vdisk VD1:

```
# trust enable
Trust Virtual-disk Enabled.

# trust vdisk VD1
Are you sure? yes
Virtual-disk VD1 has been trusted.
```

See also
- show vdisks

unmap volume

Description Deletes an explicit mapping whose settings override a volume's default mapping. When the explicit mapping is deleted, host access to the volume is controlled by the volume's default mapping (described in help for create volume).

Syntax `unmap volume [host host] volume`

Parameters

`host host`
Optional. For FC and SAS, the host's nickname or 16-hex-digit WWPN. For iSCSI, the iSCSI host initiator's node name (typically the IQN) or nickname. If this parameter is omitted, mapping changes apply to all hosts not explicitly mapped.

volume
Name or serial number of the volume to unmap. For syntax, see Command syntax.

Example Unmap volume V1 from host Host1:

```
# unmap volume host Host1 V1
Success: Command completed successfully. - The volume was unmapped successfully.
```

Unmap volume V2's default mapping (leaving explicit mappings unchanged):

```
# unmap volume V2
Success: Command completed successfully. - The volume was unmapped successfully.
```

See also
- map volume
- show host-maps
- show hosts
- show volume-maps
- show volumes

verify vdisk

Description Verifies whether vdisk redundancy data is consistent with its user data. For RAID 3, 5, 6, and 50, verify checks all parity blocks to find data-parity mismatches. For RAID 1 and 10, verify compares the primary and secondary disks to find data inconsistencies.

Verification can last over an hour, depending on vdisk size, utility priority, and amount of I/O activity. When verification is complete, the number of inconsistencies found is reported with event code 21 (Vdisk verification complete) in the event log. Such inconsistencies can indicate that a disk in the vdisk is going bad. You can use a vdisk while it is being verified.

Syntax `verify vdisk vdisks`

Parameters

vdisks
Name or serial number of the vdisks to verify. For syntax, see Command syntax.

Example Verify vdisk VD1:

```
# verify vdisk VD1
Info: Verify started on vdisk VD1
Success: Command completed successfully
```

See also
- `abort verify`
- `show vdisks`

versions

Description Shows the hardware and software versions for each controller module. Alias: `show versions`.

Syntax
```
versions
```

Example Show versions for a single-controller system:

```
# versions
Controller A Versions
---------------------
Storage Controller CPU Type: ver
Storage Controller Code Version: ver
Memory Controller FPGA Code Version: ver
Storage Controller Loader Code Version: ver
Management Controller Code Version: ver
Management Controller Loader Code Version: ver
Expander Controller Code Version: ver
CPLD Code Version: ver
Hardware Version: ver
Host Interface Module Version: ver
Host Interface Module Model: ver
```

volumecopy

Description Copies a snapshot or a master volume to a new standard volume. The command creates the destination volume you specify, which must be in a vdisk owned by the same controller as the source volume. While the copy operation is in progress, the destination volume's type is shown as `standard*`; when complete, it changes to `standard`.

Before copying a master volume, verify that the snap-pool has space for the temporary snapshot, which is used to track changes to the master volume while the copy is in progress. Also, you must unmount the master volume from hosts. After the volume copy has started, you can remount the master volume.

Before copying a snapshot volume with its modified data, you must unmount it from hosts. When the volume copy starts, the snapshot and the destination volume will be offline (unavailable to hosts) until the operation is complete.

△ **CAUTION:** Copying a mounted master volume or a mounted snapshot volume (when modified data is included) will result in data corruption.

Syntax
```
volumecopy
    source-volume volume1
    dest-vdisk vdisk
    [modified-snapshot yes|no]
    volume2
```

Parameters

`source-volume` *`volume1`*
Name or serial number of the snapshot or master volume to copy. For syntax, see Command syntax.

dest-vdisk *`vdisk`*
Name or serial number of the destination vdisk. For syntax, see Command syntax.

`modified-snapshot yes|no`
Optional. Specifies whether to include or exclude modified write data from the snapshot in the copy. This parameter applies only when the source volume is a snapshot; it is ignored if the source volume is a master volume.

- `yes`: Include modified snapshot data.
- `no`: Exclude modified snapshot data.

If this parameter is omitted for a snapshot, modified snapshot data is excluded.

`volume2`
A name for the volume to create in the destination vdisk. For syntax, see Command syntax.

Example Copy master volume `MV1` to new volume `MV1copy` on vdisk `VD2`:

```
# volumecopy source-volume MV1 dest-vdisk VD2 Copy
Leaving the source volume mounted when starting a volume copy operation will
result in data corruption. The source volume must be unmounted prior to beginning
the volume copy operation. The source volume can be remounted once the volume
copy has started.
In addition, once volume copy starts, the destination volume will be created, and
will be offline until the volume copy operation is complete.
Is the source volume unmounted from all Operating Systems? yes
Success: Command completed successfully. - The volume copy started.

# show volumes
Vdisk  Name      ...  Type
-------------------------------------------------------------------
VD2    MV1       ...  mastervol
VD2    MV1copy   ...  standard*
-------------------------------------------------------------------
```

See also

- abort volumecopy
- create task
- show vdisks
- show volumecopy-status
- show volumes

Glossary

CHAP	Challenge-Handshake Authentication Protocol.
chunk size	The amount of contiguous data that is written to a vdisk member before moving to the next member of the vdisk.
dedicated spare	A disk that is reserved for use by a specific vdisk to replace a failed disk.
default mapping	Host-access settings that are configured when a volume is created.
dual-port disk	A dual-port disk is connected to both controllers so its data path is fault tolerant.
dynamic spare	A properly sized available disk that is automatically assigned, if the dynamic spares option is enabled, to replace a failed disk in a redundant vdisk.
EC	Expander Controller. The processor (located in the SAS expander in each controller module and expansion module) that is primarily responsible for enclosure management and SES.
EMP	Enclosure management processor. An EC subsystem that provides SES data such as temperature, power supply and fan status, and the presence or absence of disks.
FC	Fibre Channel interface protocol.
IQN	iSCSI Qualified Name.
iSCSI	Internet SCSI interface protocol.
iSNS	Internet Storage Name Service.
jumbo frame	In an iSCSI network, a frame that can contain 9000 bytes for large data transfers. A normal frame can contain 1500 bytes.
global spare	A disk that is reserved for use by any redundant vdisk to replace a failed disk.
host	An external port that the storage system is attached to. The external port may be a port in an I/O adapter in a server, or a port in a network switch.
leftover	The state of a disk when its metadata says the disk is a member of a vdisk but other members' metadata say the disk isn't a member. The metadata must be cleared before the disk can be used in a new vdisk or as a spare.
loop	Fibre Channel Arbitrated Loop (FC-AL) topology.
masking	Volume-mapping settings that specify no access to that volume by hosts.
master volume	A volume that is enabled for snapshots and has an associated snap pool.
MC	Management Controller. The processor (located in a controller module) that is primarily responsible for human-computer interface and computer-computer interface functions, and interacts with the SC.
metadata	Data in the first sectors of a disk drive that stores all disk, vdisk, and volume specific information including vdisk membership or spare identification, vdisk ownership, volumes and snapshots in the vdisk, host mapping of volumes, and results of the last media scrub.
point-to-point	Fibre Channel Point-to-Point topology.
SAS	Serial Attached SCSI interface protocol or disk-drive architecture.
SATA	Serial ATA disk-drive architecture.
SC	Storage Controller. The processor (located in a controller module) that is primarily responsible for RAID controller functions. The SC is also referred to as the RAID controller.
secret	For use with CHAP, a password that is shared between an initiator and a target to enable authentication.
SES	SCSI Enclosure Services.

single-port disk	A single-port disk is connected to both controllers so its data path is not fault tolerant. A single-port disk's type is shown as SAS-S or SATA-S.
snap pool	A volume that stores data that is specific to snapshots of an associated master volume, including copy-on-write data and data written explicitly to the snapshots. A snap pool cannot be mapped.
snapshot	A “virtual” volume that preserves the state of a master volume’s data as it existed when the snapshot was created. Data associated with a snapshot is recorded in both the master volume and in its associated snap pool. A snapshot can be mapped and written to.
SSD	Solid-state drive. Supported SSDs have a SATA interface.
unwritable cache data	Data that has not been written to disk and is associated with a volume that no longer exists or whose disks are not online. If the data is needed, the volume's disks must be brought online. If the data is not needed it can be cleared, in which case it will be lost and data will differ between the host and disk. Unwritable cache is also called orphan data.
vdisk	A “virtual” disk comprising the capacity of one or more disks. The number of disks that a vdisk can contain is determined by its RAID level.
volume	A portion of the capacity of a vdisk that can be presented as a storage device to a host.
volume copy	Licensed capability to create a copy, or clone, a volume.
ULP	Unified LUN Presentation. A RAID controller feature that enables a host to access mapped volumes through any controller host port.
WWN	World Wide Name. A globally unique 64-bit number that identifies a node process or node port.
WWNN	World Wide Node Name. A globally unique 64-bit number that identifies a node process.
WWPN	World Wide Port Name. A globally unique 64-bit number that identifies a node port.

Index

D

E

F

G

P

Q

R

S

T

U

V

W

X