IBM Spectrum Virtualize

Interfacing Using the RESTful API

Note

Before using this information and the product it supports, read the information in "Notices" on page 785.

This edition applies to version 8, release 1, modification 3, and to all subsequent modifications until otherwise indicated in new editions.

Contents

Chapter 9. Clustered system diagnostic and service-aid commands 317

Chapter 10. Controller commands . . 351

Chapter 11. Drive commands. 359

Chapter 12. Email and event notification commands 379

Chapter 13. Enclosure commands . . 397

Chapter 14. Encryption commands 433

Chapter 15. Licensing and featurization commands. 447

Chapter 16. FlashCopy commands 457

Chapter 17. Host commands 483

Chapter 18. Information commands 511

Chapter 19. Livedump commands . . 513

Tables

About this guide

This publication provides information that helps you configure and use the IBM Spectrum Virtualize™ for SAN Volume Controller and Storwize® Family.

Accessibility

IBM® strives to provide products with usable access for everyone, regardless of age or ability.

This product uses standard Windows navigation keys.

For more information, see the accessibility features topic in the "Reference" section.

Emphasis

Different typefaces are used in this guide to show emphasis.

The following typefaces are used to show emphasis.

Emphasis	Meaning
Boldface	Text in **boldface** represents menu items.
`Bold monospace`	Text in **`bold monospace`** represents command names.
Italics	Text in *italics* is used to emphasize a word. In command syntax, it is used for variables for which you supply actual values, such as a default directory or the name of a system.
`Monospace`	Text in `monospace` identifies the data or commands that you type, samples of command output, examples of program code or messages from the system, or names of command flags, parameters, arguments, and name-value pairs.

Syntax diagrams

A syntax diagram uses symbols to represent the elements of a command and to specify the rules for using these elements.

Table 1 on page xii explains how to read the syntax diagrams that represent the command-line interface (CLI) commands. In doing so, it defines the symbols that represent the CLI command elements.

Table 1. Syntax diagrams

Element	Syntax	Description
Main path line	>>><>() () ()	The main path line begins on the left with double arrowheads: >>. The main path line ends on the right with two arrowheads facing each other: ><. If a diagram is longer than one line, each line to be continued ends with a single arrowhead and the next line begins with a single arrowhead: > Read the diagrams from left-to-right, top-to-bottom, following the main path line.
Keyword	►►──esscli──────────────────►◄	Represents the name of a command, flag, parameter, or argument. A keyword is not in italics. Spell a keyword exactly as it is shown in the syntax diagram.
Required keywords	►►─┬─a─*AccessFile*─┬──────────►◄ ├─u─*Userid*─────┤ └─p─*Password*───┘	Indicate the parameters or arguments that you must specify for the command. Required keywords must be written on the main path line. Required keywords that cannot be used together are stacked vertically.
Optional keywords	►►──┬──────┬──────────►◄ ├─h─┤ ├─-help─┤ └─?─┘	Indicate the parameters or arguments that you can choose to specify for the command. Optional keywords must be written below the main path line. Mutually exclusive optional keywords are stacked vertically.
Default value	┌─FCP───┐ ►►──protocol──=──┴─FICON─┴──────►◄	The default value must be written above the main path line.
Repeatable keyword or value	►►──newports──=──┬─ALL──────────────┬──►◄ └─*PortId1,PortId2,...*─┘	Represents a parameter or argument that you can specify more than once. A repeatable keyword or value is represented by an arrow returning to the left above the keyword or value.
Variable	►►──*AccessFile*──────────────►◄	Represents the value that you need to supply for a parameter or argument, such as a file name, user name, or password. Variables are in italics.
Space separator	►►──u── ──*Userid*── ──p── ──*Password*──────►◄	Adds a blank space on the main path line to separate keywords, parameters, arguments, or variables from each other.

Table 1. Syntax diagrams (continued)

Element	Syntax	Description
Quotation mark delimiters	►►—d— —"—ess—=—*EssId*—host—=——————► ►—*'Host Name'*— —profile—=—*ProfileName*———► ►—"————————————————————►◄	Indicates the start and end of a parameter or argument that contains multiple values. Enclose one or more name-value pairs in a set of double quotation marks for a particular parameter or argument. If the value of a parameter or name-value pair contains a blank or white space, enclose the entire value in a set of single quotation marks.
Equalsign operator	►►—"—ess—=—*EssId*— —profile—=——————► ►—*ProfileName*—"———————————————►◄	Separates a name from its value in a name-value pair.
Syntax fragment	►►—Fragment Name———————————————►◄ **Fragment name:** ├—(—*fragment details*—)——————————┤	Breaks up syntax diagrams that are too long, too complex, or repetitious. The fragment name is inserted in the main diagram, and the actual fragment is shown below the main diagram.

CLI special characters

The following special characters are used in the command-line interface (CLI) command examples.

minus (-) sign
Flags are prefixed with a - (minus) sign. Flags define the action of a command or modify the operation of a command. You can use multiple flags, followed by parameters, when you issue a command. The - character cannot be used as the first character of an object name.

vertical bar (|)
A vertical bar signifies that you choose only one value. For example, [a | b] in brackets indicates that you can choose a, b, or nothing. Similarly, { a | b } in braces indicates that you must choose either a or b.

delimiters (: or , or !)
Delimiters are used to delimit items listed after issuing an information command.

- Colon (:) is used to delimit items in a list in a command (for example `mkhost -name myhost -hbawwpn AA22000011112222:AA22000011112223`).
- Comma (,) is used to delimit items in a list in a command if item values can contain a colon.
- Exclamation mark (!) is used to delimit items in a command if the item values can contain a colon or comma. Exclamation points generally do not show up in example output and are a good delimiter to use.

Using wildcards in the CLI

You can use wildcards in the system command-line interface (CLI).

The CLI supports the use of the asterisk character (*) as a wildcard within the arguments of certain parameters. There are some behavioral issues that must be considered when using wildcards in order to prevent unexpected results. These behavioral issues and the ways to avoid them are as follows:

1. Running the command while logged onto the node.

The shell will attempt to interpret any of the special characters if they are not escaped (preceded with a backslash character). Wildcards will be expanded into a list of files if any files exist that match the wildcards. If no matching files exist, the wildcard is passed to the system command untouched.

To prevent expansion, issue the following command in one of its formats:

cleardumps -prefix '/dumps/*.txt' with single quotation marks (' '), or

cleardumps -prefix /dumps/*.txt using a backslash (\), or

cleardumps -prefix "/dumps/*.txt" with double quotation marks (" ").

2. Running the command through Secure Shell (SSH), for example from a host.

This method is slightly more complicated because the host shell processes the command line before it is passed through SSH to the shell on the clustered system (system). This means an extra layer of protection is required around the wildcard as the host shell will strip off any protecting quotes, and if the wildcard is exposed to the system shell, this will result in the wildcard being expanded in the system shell.

To prevent expansion, issue the following command in one of its formats:

cleardumps "'/dumps/*.txt'" with single quotation marks (' ') inside of double quotation marks (" "), or

cleardumps '/dumps/*.txt' using a backslash (\) inside of single quotation marks (' '), or

cleardumps '"/dumps/*.txt"' with double quotation marks (" ") inside of single quotation marks (' ').

Data types and value ranges

The maximum length of any single parameter entered into the command line is 2176 bytes.

Note: When creating a new object, the clustered system (system) assigns a default **-type** name if one is not specified. The default **-type** name consists of the object prefix and the lowest available integer starting from 0 (except for nodes starting from 1); for example, *vdisk23*; the default -type name must be unique.

Table 2 lists the data types and the value ranges for each.

Table 2. Data types

Data types	Value ranges
filename_arg	This is a (optionally fully qualified) file name, containing a maximum of 169 characters. Valid characters are: • . (period; the field must not start with, end with, or contain two consecutive periods) • / (forward slash) • - (hyphen) • _ (underscore) • a–z (lowercase letters, A through Z) • A–Z (uppercase letters, A through Z) • 0–9 (numerals 0 through 9)

Table 2. Data types (continued)

Data types	Value ranges
`directory_or_file_filter`	Specifies a directory, file name filter, or both, within the specified directory. Valid directory values are: • `/dumps` • `/dumps/audit` • `/dumps/configs` • `/dumps/elogs` • `/dumps/feature` • `/dumps/iostats` • `/dumps/iotrace` • `/dumps/software` The file name filter can be any valid file name, containing a maximum of 128 characters, with or without the "*" (wildcard), and appended to the end of a directory value. Valid characters are: • * (asterisk/wildcard) • . (the field must not start with, end with, or contain two consecutive periods) • / • - • _ • a–z • A–Z • 0–9
`filename_prefix`	The prefix of a file name, containing a maximum of 128 characters. Valid characters are: • - • _ • a–z • A–Z • 0–9
`name_arg`	Names can be specified or changed using the **create** and **modify** functions. The view commands provide the name and ID of an object. **Note:** The system name is set when the system is created. The first character of a *name_arg* must be nonnumeric. The first character of an object name cannot be a – (dash) because the CLI (command-line interface) interprets it as being the next parameter. Valid characters are: • **.** (a period - the field must not start with, end with, or contain two consecutive periods) • / • - • _ • space • `a` through `z` • `A` through `Z` • `0` through `9`

Table 2. Data types (continued)

Data types	Value ranges
`password`	This is a user-defined password containing a maximum of 15 characters. Valid characters are: • - (cannot be used as the first character) • _ • a–z • A–Z • 0–9
`serial_number`	The format of this number conforms to IBM standard C-S 1-1121-018 1999-06 Serial Numbering for IBM products. The serial number is 7 digits, the first two of which define the manufacturing location, leaving 5 digits for the product. The standard defines a way to extend the serial number using letters in the place of numbers in the 5-digit field.
`ip_address_arg`	The argument follows the standard rules for dotted decimal notation. The following Internet Protocol 4 (IPv4) and Internet Protocol 6 (IPv6) address formats are supported: **IPv4 (no port set, SAN Volume Controller uses default)** 1.2.3.4 **IPv4 with specific port** 1.2.3.4:22 **Full IPv6, default port** 1234:1234:0001:0123:1234:1234:1234:1234 **Full IPv6, default port, leading zeros suppressed** 1234:1234:1:123:1234:1234:1234:1234 **Full IPv6 with port** [2002:914:fc12:848:209:6bff:fe8c:4ff6]:23 **Zero-compressed IPv6, default port** 2002::4ff6 **Zero-compressed IPv6 with port** [2002::4ff6]:23
`dns_name`	This is the dotted domain name for the system subnet (for example, yourcompany.com).
`hostname`	The host name assigned to the system. This name can be different from the system name, and is modifiable. A combination of the host name and the *dns_name* is used to access the system, for example: https://hostname.yourcompany.com
`capacity_value`	The capacity expressed within a range of 512 bytes to 2 petabytes (PB). **Tip:** Specify the capacity as megabytes (MB), kilobytes (KB), gigabytes (GB), or PB. When using MB, specify the value in multiples of 512 bytes. A capacity of 0 is valid for a striped or sequential volume. The smallest number of supported bytes is 512.
`node_id`	A node ID differs from other IDs in that it is a unique ID assigned when a node is used to create a system, or when a node is added to a system. A *node_id* value is never reused in a system. Node IDs are internally represented as 64-bit numbers, and like other IDs, cannot be modified by user commands.

Table 2. Data types (continued)

Data types	Value ranges
`xxx_id`	All objects are referred to by unique integer IDs, assigned by the system when the objects are created. All IDs are represented internally as 32-bit integers; node IDs are an exception. IDs in the following ranges identify the various types of objects: • node_id: A positive decimal integer greater than or equal to 1 • mdisk_grp_id: 0–127 • io_grp_id: 0–3 (See Note.) • mdisk_id: 0–4095 • vdisk_id: 0–8191 • copy_id: 0–1 • host_id: 0–1023 • flash_const_grp_id: 0–255 • remote_const_grp_id: 0–255 • fcmap_id: 0–4095 • rcrel_id: 0–8191 • controller_id: 0–63 **Note:** The io_group 4 exists but is used only in certain error recovery procedures. These IDs, like node IDs, cannot be modified by user commands. **Note:** IDs are assigned at run time by the system and cannot be relied upon to be the same after; for example, the configuration restoration. Use object names in preference to IDs when working with objects.
`xxx_list`	A colon-delimited list of values of type *xxx*.
`wwpn_arg`	The Fibre Channel worldwide port name (WWPN), expressed as a 64-bit hexadecimal number and consisting of the characters 0–9, a–f, and A–F; for example: `1A2B30C67AFFE47B`. **Note:** Entering `WWPN 0` in the command string causes a command failure.
`panel_name`	This is a string of up to six characters corresponding to the number on the printed label below the display on the front panel of a node in the system.
`sequence_number`	A 32-bit unsigned integer, expressed in decimal format.
`csi_num_arg`	A 32-bit unsigned integer, expressed in decimal format.
`percentage_arg`	An 8-bit unsigned integer, expressed in decimal 0–100 format.
`extent_arg`	A 32-bit unsigned integer, expressed in decimal format.
`num_extents_arg`	A 32-bit unsigned integer, expressed in decimal format.
`threads_arg`	An 8-bit unsigned integer, expressed in decimal format. Valid values are 1, 2, 3, or 4.
`velocity_arg`	The fabric speed in gigabytes per second (GBps). Valid values are 1 or 2.
`timezone_arg`	The ID as detailed in the output of the **`lstimezones`** command.
`timeout_arg`	The command timeout period. An integer from 0 to 600 (seconds).
`stats_time_arg`	The frequency at which statistics are gathered. Valid values are 1 to 60 minutes in increments of 1 minute.

Table 2. Data types (continued)

Data types	Value ranges
`directory_arg`	Specifies a directory, file name filter, or both, within the specified directory. Valid directory values are: • `/dumps` • `/dumps/audit` • `/dumps/cimom` • `/dumps/configs` • `/dumps/elogs` • `/dumps/feature` • `/dumps/iostats` • `/dumps/iotrace` • `/home/admin/upgrade` The file name filter can be any valid file name, containing a maximum of 128 characters, with or without the wildcard (*, an asterisk), and appended to the end of a directory value. Valid characters are: • * • . (the field must not start with, end with, or contain two consecutive periods) • / • - • _ • a–z • A–Z • 0–9
`locale_arg`	The system locale setting. Valid values are: • 0 en_US: US English (default) • 1 zh_CN: Simplified Chinese • 2 zh_TW: Traditional Chinese • 3 ja_JP: Japanese • 4 fr_FR: French • 5 de_DE: German • 6 it_IT: Italian • 7 es_ES: Spanish
`key_arg`	A user-defined identifier for a secure shell (SSH) key, containing a maximum of 30 characters.
`user_arg`	Specifies the user: admin or service.
`copy_rate`	A numeric value of 0–100.
`copy_type`	Specifies the Mirror copy type: Metro or Global.

The maximum number of values entered into a colon-separated list is 128; exceeding this maximum number returns an error.

CLI commands and parameters

Command-line interface (CLI) commands and parameters are represented in the syntax diagram.

The system command-line interface offers command line completion for command entry. Command line completion allows you to type in the first few characters of a command and press the Tab key to fill in the rest of the command name. If there are multiple commands that start with the same characters, then a list of possible commands is returned. You can type in more characters until the command name is unambiguous.

CLI parameters can be entered in any order except in the following situations:

- When a command name is specified, the first argument given must be the action that you want to be performed.
- Where you are specifying a command against a specific object, the object ID or name must be the last argument in the line.

A valid parameter meets the following requirements:

- Parameters can be entered in any order.
- If a parameter has an associated argument, the argument must always follow the parameter.
- A parameter must start with a '-'; otherwise, it is assumed to be an argument.
- The maximum length of any single parameter that can be entered into the CLI is 128 bytes.
- An argument can contain multiple data items. The maximum number of data items that you can enter into such a list is 128. For a component list, separate the individual items by a colon.
- Any parameter with an argument can be entered as `-parameter=argument`.
- Entering `-param=` means the argument is an empty string, equivalent to `-param`.
- The symbol '--' is valid as the next to last entry on the command line. It specifies that the next entry is the target object name or ID, even if it begins with a hyphen.

  ```
  chuser -usergrp=-usergrp -- -password
  ```
- The symbol '--' is valid as the final word on the command line.

Examples that are valid

```
mkuser -name fred -usergrp 0 -password buckets
mkuser -name fred -usergrp 0 -password=buckets
mkuser -name=-barney -usergrp=0 -password=buckets

chuser -usergrp 1 fred
chuser -usergrp 1 -- fred
chuser -usergrp 1 -- -barney
```

Examples that are invalid

```
chuser -usergrp 1 fred --
chuser -usergrp 1 -- fred --
chuser -- -usergrp 1 fred
chuser -usergrp 1 -barney
```

CLI flags

The following flags are common to all command-line interface (CLI) commands.

-? or -h

Print help text. For example, issuing **`lssystem -h`** provides a list of the actions available with the **`lssystem`** command.

-nomsg

When used, this flag prevents the display of the `successfully created` output. For example, if you issue the following command:

```
mkmdiskgrp -ext 16
```

it displays:

```
MDisk Group, id [6], successfully created
```

However, if the -nomsg parameter is added, for example:

```
mkmdiskgrp -ext 16 -nomsg
```

the following information is displayed:

```
6
```

This parameter can be entered for any command, but is only acted upon by those commands that generate the `successfully created` outputs. All other commands ignore this parameter.

CLI messages

Ensure that you are familiar with the command-line interface (CLI) messages.

When some commands complete successfully, textual output is normally provided. However, some commands do not provide any output. The phrase `No feedback` is used to indicate that no output is provided. If the command does not complete successfully, an error is generated. For example, if the command has failed as a result of the cluster being unstable, the following output is provided:

- `CMMVC5786E The action failed because the cluster is not in a stable state.`

The Knowledge Center has a description of each message you might see.

Understanding capacity indicators

The system uses base-2 (binary numeral) as capacity indicators for volumes, drives, and other system objects. The management GUI and the command-line interface (CLI) use different abbreviations to indicate capacity.

The following table displays the differences in how capacity indicators are displayed in the management GUI and the CLI.

Table 3. Capacity indicators. This table displays the differences in how capacity indicators are displayed in the management GUI and the CLI.

Metric	GUI Abbreviation	CLI Abbreviation	Value
kibibyte	KiB	KB	1024
mebibyte	MiB	MB	1,048,576
gibibyte	GiB	GB	1,073,741,824
tebibyte	TiB	TB	1,099,511,627,776
pebibyte	PiB	PB	1,125,899,906,842,624
exbibyte	EiB	EB	1,152,921,504,606,846,976
zebibyte	ZiB	ZB	1,180,591,620,717,411,303,424
yobibyte	YiB	YB	1,208,925,819,614,629,174,706,176

Attributes of the -filtervalue parameters

The **`-filtervalue`** parameter filters a view that is based on specific attribute values that relate to each object type. You can combine multiple filters to create specific searches, for example, **`-filtervalue name=fred:status=online`**. The help (**`-filtervalue`**) specifies the attributes that are available for each object type.

The **`-filtervalue`** parameter must be specified with *attrib=value*. The **`-filtervalue?`** and **`-filtervalue`** parameters cannot be specified together.

Note: The qualifier characters less than (<) and greater than (>) must be enclosed within double quotation marks (""). For example, **-filtervalue vdisk_count "<"4 or port_count ">"1**. It is also valid to include the entire expression within double quotation marks. For example, **-filtervalue "vdisk_count<4"** .

When an attribute requires the **-unit** parameter, it is specified after the attribute. For example, **-filtervalue capacity=24 -unit mb**. The following input options are valid for the **-unit** parameter:

- b (bytes)
- kb (Kilobytes)
- mb (Megabytes)
- gb (Gigabytes)
- tb (Terabytes)
- pb (Petabytes)

Capacity values displayed in units other than bytes might be rounded. When filtering on capacity, use a unit of bytes, **-unit b**, for exact filtering.

You can use the asterisk (*) character as a wildcard character when names are used. The asterisk character can be used either at the beginning or the end of a text string, but not both. Only one asterisk character can be used in a **-filtervalue** parameter.

Chapter 1. Spectrum Virtualize RESTful API

The Spectrum Virtualize Representational State Transfer (REST) model Application Programming Interface (API) consists of command targets that are used to retrieve system information and to create, modify, and delete system resources. These command targets allow command parameters to pass through unedited to the Spectrum Virtualize command line interface, which handles parsing parameter specifications for validity and error reporting. Use Hypertext Transfer Protocol Secure (HTTPS) to successfully communicate with the RESTful API server.

The RESTful API server does not consider transport security (such as SSL), but instead assumes that requests are initiated from a local, secured server. The HTTPS protocol provides privacy through data encryption. The RESTful API provides more security by requiring command authentication, which persists for two hours of activity, or 30 minutes of inactivity, whichever occurs first.

Uniform Resource Locators (URLs) target different node objects on the system. The HTTPS POST method acts on command targets that are specified in the URL. For more information, see "RESTful API command targets and characteristics" on page 2. To make changes or view information about different objects on the system, you must create and send a request to the system. You are required to provide certain elements for the RESTful API server to receive and translate the request into a command, as described in the next section.

Making an HTTPS request

To interact with the system by using the RESTful API, make an HTTPS command request with a valid configuration node URL destination. Include the 7443 port and the keyword rest. Use the following URL format for all requests:

```
https://system_node_ip:7443/rest/command
```

Where:

- *system_node_ip* is the system IP address, which is the address that is taken by the configuration node of the system.
- The port number is always 7443 for the Spectrum Virtualize RESTful API.
- rest is a keyword.
- *command* is the target command object (such as **auth** or **lseventlog** with any parameters).

  The *command* specification follows this format:

  ```
  command_name,method="POST",headers={'parameter_name': 'parameter_value',
  'parameter_name': 'parameter_value',...}
  ```

Why we use POST exclusively: All of the Spectrum Virtualize RESTful API command targets are named after Spectrum Virtualize commands, with names that already reflect Create, Read, Update, and Delete actions. MK commands make (create) resources, LS commands list (read) resources, CH commands change (update) resources, and RM commands remove (delete) resources. Using HTTP methods on top of such commands is redundant.

All commands, including LS commands, accept at least one named parameter. Because the Spectrum Virtualize RESTful API is POST method only, it implements a convention of appending positional parameters to the URI, and packing named parameters into the request body as JavaScript Object Notation (JSON) strings.

Why we do not use GET: Most HTTP servers reject GET requests with body data. To support GET would mean appending named parameters to the URI as a query string. Do not include such arbitrary data in a query string. A valid URI cannot contain whitespace and other reserved characters that might be present in the data. Therefore, you must URL-encode named parameter data before you append it to the URI. In addition to that inconvenience, the key-value structure of named parameter data must be conveyed in some way. The query string would likely be a URL-encoded JSON object. Supporting the GET method means that parameter data represented in your programming language of choice must pass through two separate encoders before the data is sent as part of an HTTP request.

The more elegant method for the command targets was to adopt POST exclusively.

As mentioned, in addition to the URL and the name of the command target, other information is required in the request line and in the body of the HTTP request that regards the action to take on the specified object. In the request line, include the POST HTTP method. Include any required parameters (such as RAID level or IP address) in the body of the request.

Provide any required parameters as valid JSON in the HTTP body, such as shown in the following example:

```
{'X-Auth-Username': 'superuser'}
```

The request is routed to port 7443 on the specified destination (which must be the configuration node of the system), where the request is received by the RESTful API server. The server runs the command, collects any resulting output, then creates an HTTP response like the following example:

```
HTTP/1.1 200 OK
Server: lighttpd/1.4.31
Date: date
Content-type: application/json; charset=UTF-8
Content-length: content_length
Connection: close
{"attribute": "value"}
```

To view API command targets and their characteristics, see “RESTful API command targets and characteristics.” To see an example of how to get started, see “Getting started” on page 4. See Appendix C, “HTTP error messages,” on page 777 for a complete list of HTTP error codes that you might encounter.

RESTful API command targets and characteristics

Table 4 stresses the POST method for all commands, including /auth. It also shows that you must use the authentication token that is returned by the /auth command target to authenticate every other command you run. Except for the /auth command target, you run commands against the system IP address so that they are run by the configuration node.

Table 4. POST method, authentication requirements, and whether to run on the configuration node

Command targets	Method	Authentication required	Run in Configuration Node / Cluster
/auth	POST	No	No
All other command targets	POST	Yes	Yes

Table 5 on page 3 shows the command target names of commonly used commands for this release of the RESTful API. Following convention, the svcinfo and svctask executable commands are defaults and do not require listing within your RESTful API command targets.

Descriptions of the command targets and their parameters, and descriptions of other less frequently used commands are available in the CLI command section of the product documentation.

Table 5. Supported RESTful API commands for Spectrum Virtualize software

Command targets		
/addhostclustermember	/addhostiogrp	/addhostport
/addvdiskaccess	/addvdiskcopy	/addvolumecopy
/auth	/chhost	/chnode
/chnodecanister	/chrcconsistgrp	/chrcrelationship
/chvdisk	/expandvdisksize	/lscurrentuser
/lseventlog	/lsfcconsistgrp	/lsfcmap
/lsfcmapcandidate	/lsfcmapdependentmaps	/lsfcmapprogress
/lshost	/lshostcluster	/lshostclustermember
/lshostclustervolumemap	/lshostiogrp	/lshostvdiskmap
/lsiogrp	/lsiogrphost	/lsmdiskgrp
/lsnode	/lsnodecanister	/lsnodehw
/lsnodecanisterhw	/lsnodecanisterstats	/lsnodecanistervpd
/lsnodehw	/lsnodestats	/lsnodevpd
/lspartnership	/lsrcrelationshipprogress	/lssystem
/lssystemip	/lssystemstats	/lsvdisk
/lsvdiskaccess	/lsvdiskcopy	/lsvdiskfcmapcopies
/lsvdiskfcmappings	/lsvdiskhostmap	/lsvdisksyncprogress
/mkfcconsistgrp	/mkfcmap	/mkfcpartnership
/mkhost	/mkhostcluster	/mkrcconsistgrp
/mkrcrelationship	/mkvdisk	/mkvdiskhostmap
/mkvolume	/mkvolumehostclustermap	/movevdisk
/prestartfcconsistgrp	/prestartfcmap	/rmfcmap
/rmhost	/rmhostcluster	/rmhostclustermember
/rmhostiogrp	/rmhostport	/rmvdisk
/rmvdiskaccess	/rmvdiskcopy	/rmvdiskhostmap
/rmvolume	/rmvolumecopy	/rmvolumehostclustermap
/startfcmap	/startrcconsistgrp	/startrcrelationship
/stopfcconsistgrp	/stopfcmap	/stoprcconsistgrp
/stoprcrelationship		

Authentication overview

Aside from data encryption, the HTTPS server requires authentication of a valid user name and password for each API session. Use two authentication header fields to specify your credentials: X-Auth-Username and X-Auth-Password.

Initial authentication requires that you POST the authentication target (/auth) with the user name and password. The RESTful API server returns a hexadecimal token. A single session lasts a maximum of two active hours or thirty inactive minutes, whichever occurs first. When your session ends due to inactivity, or if you reach the maximum time that is allotted, error code 403 indicates the loss of authorization. Use the /auth command target to reauthenticate with the user name and password.

For example, the following command passes the authentication command to target node IP 192.168.10.109 at port 7443:

```
https://192.168.10.109:7443/rest/auth, method="POST",
   headers={'X-Auth-Username': 'superuser', 'X-Auth-Password': 'passw0rd'}
```

The HTTP request that was sent to the API server looks like:

```
POST /auth HTTPS/HTTPS_version
Host: https://192.168.10.109:7443
Content-type: application/json; charset=UTF-8
Content-length: message_size
X-Auth-Token: 58cfd6acb1676e1cba78b7cb5a9a081d11d1d1cfeb0078083ef225d9c59bf4df

{"attribute": "value","attribute": "value"}
```

Where:

- The first line is the request line with the POST method, API target, protocol (HTTPS), and protocol version (1.1).
- The second line is the host header, directing the HTTP request to the correct port (7443) and IP address on the system.
- The third line is the content type header that specifies the content type (application/json; charset=UTF-8).
- The fourth line is the content length header with the message size.
- The fifth line is the authentication token header, with the authentication token.
- A space is left between the headers and the body of the request. Any parameters appear in JSON on the seventh line.

The successful **auth** command returns a token similar to the following example:

```
{
"token": "58cfd6acb1676e1cba78b7cb5a9a081d11d1d1cfeb0078083ef225d9c59bf4df"
}
```

Command target parameters

Command targets interact with different parts of the system. After you target a specific node in the system (usually a configuration node), target an object within that node. See Table 5 on page 3 for command targets and see the CLI commands section of this product documentation for individual command descriptions of parameters that you can specify.

Getting started

The following Python 3 example shows how to complete initial setup to start interacting with the system and running commands. For examples in other languages, see Appendix D, “Usage examples in Perl,” on page 779 and Appendix E, “Usage examples in CURL,” on page 781.

```
import ssl
import json
import pprint

import urllib.request
import urllib.error
import urllib.parse

no_verify = ssl.create_default_context()
no_verify.check_hostname = False
no_verify.verify_mode = ssl.CERT_NONE

if getattr(ssl, '_https_verify_certificates', None):
    ssl._https_verify_certificates(False)
```

```
class HostString(str):
    """
        Comment: Special subclass of string, for storing arbitrary host-related
                 attributes (such as auth tokens) without losing any string behavior
    """
    def __new__(cls, *args, **kwds):
        return super(HostString, cls).__new__(cls, *args, **kwds)

class RESTUtil(object):
    show_default=False
    default_headers = {}
    port = 80

    def __init__(self, show=None, catch=True):
        self.hosts = {}
        self.curr_host = None
        self.catch=catch
        self.show_default=show if show != None else self.show_default

    @property
    def host(self):
        return self.curr_host

    @host.setter
    def host(self, hostname):
        """
            Comment: Retrieve the HostString object of a known host from its
                     host name or string definition. Even if the host definition
                     is provided, we still need to key into self.hosts in case the
                     client classes are storing things on their HostString objects.
        """
        try:
            if hostname in self.hosts:
                self.curr_host = self.hosts[hostname]
            else:
                self.curr_host = [h for h in self.hosts.values() if h == hostname][0]
            return self.curr_host
        except IndexError:
            raise KeyError("Unrecognized host/name %s" % hostname)

    def add_host(self, hostdef, hostname=None):
        hostname = hostname if hostname is None else hostdef
        self.hosts[hostname] = HostString(hostdef)
        if self.curr_host == None:
            self.curr_host = self.hosts[hostname]
        return hostname

    def command(self, protocol, postfix, method='POST', headers=None, show=None, **cmd_kwds):
        """
            Comment: A fairly generic RESTful API request builder.
                     See subclasses for examples of use.
        """
        if show == None:
            show = self.show_default
        headers = {} if headers == None else headers
        url = '%s://%s:%s/%s' % (
            protocol,
            self.curr_host,
            self.port,
            postfix
        )
        request = urllib.request.Request(
            url,
            headers =dict(self.default_headers, **headers),
            data=bytes(json.dumps(cmd_kwds), encoding="utf-8") if cmd_kwds else None)
        request.get_method = lambda: method
        if show:
            self.request_pprint(request)
        try:
            cmd_out = urllib.request.urlopen(request, context=no_verify).read().decode('utf-8')
        except urllib.error.HTTPError as e:
            self.exception_pprint(e)
            if not self.catch:
                raise Exception("RESTful API command failed.")
            return
        try:
            cmd_out = json.loads(cmd_out)
        except ValueError:
            pass
        if show:
            print("\nCommand Output:")
            pprint.pprint(cmd_out)
            print("")
        return cmd_out

    @staticmethod
    def request_pprint(request):
        """
            Comment: Request info print function
                     (for self.command with show=True)
        """
        print(request.get_method(), request.get_full_url(), 'HTTP/1.1')
        print('Host:', request.host)
        for key, value in request.headers.items():
            print(key.upper() + ':', str(value))
        if request.data != None:
            print()
            pprint.pprint(request.data)

    @staticmethod
    def exception_pprint(http_error):
        """
            Comment: HTTPError info print function
        """
        print(http_error.code, '--', http_error.reason)
        print(http_error.fp.read())
        print("")

class SVCREST(RESTUtil):
    """
        Comment: RESTful wrapper for the SVC CLI
    """

    def __init__(self, host, *args, **kwds):
        self.debug = kwds.pop('debug', False)
        super().__init__(*args, **kwds)
        self.add_host(host)

    @property
    def default_headers(self):
        return {'X-Auth-Token': getattr(self.curr_host, 'token', 'badtoken'),
                'Content-Type': 'application/json'}

    @property
    def port(self):
        return getattr(self, '_port', None) or ('7665' if self.debug else '7443')

    @property
    def protocol(self):
        return getattr(self, '_protocol', None) or ('http' if self.debug else 'https')

    def command(self, cmd, *args, method="POST", headers=None, show=None, **cmd_kwds):
        postfix = '/'.join(
            ['rest'] + [cmd] + [urllib.parse.quote(str(a)) for a in args]
        )
        return super().command(
            self.protocol,
            postfix,
            method=method,
            headers=headers,
            show=show,
            **cmd_kwds
        )

    def authenticate(self, username='superuser', password='passw0rd', show=None):
        cmd_out = self.command(
            'auth', show=show, method="POST", headers={'X-Auth-Username': username, 'X-Auth-Password': password}
        )
        if cmd_out:
            self.curr_host.token = cmd_out['token']
    """
       Comment:  First, set your cluster ipaddress.
          It's assumed superuser/passw0rd (6 lines above) is the crednetial.
          After the authenticate call, you can issue any command in
                s.command('') that is an svcinfo or svctask cmmand)
   """
s = SVCREST('192.168.10.109')
s.authenticate()
print(s.command('lssystem'))
```

Chapter 2. API management

To authenticate, use the Superuser ID and the authentication (auth) target.

Authentication

Use the authentication command target (auth) to authenticate using the POST method. In this process, you provide a username and password through the X-Auth-Username and X-AuthPassword header fields. The beginning of the session is the only instance where you must enter the username and password. Upon successfully entering the username and password, an authentication token is displayed. This token is a 32-digit, hexadecimal code that you include in future commands using the X-Auth-Token header. The X-Auth-Token header in combination with the authentication token replaces the username and password for each action. The token is good for one session, but after two hours of activity or thirty minutes of inactivity, the session times out. Repeat authentication for another token. For information on system authentication, see “Authentication overview” on page 3.

Chapter 3. Using the CLI

The command-line interface (CLI) is a collection of commands that you can use to manage the clustered system (system).

Overview

The CLI commands use the Secure Shell (SSH) connection between the SSH client software on the host system and the SSH server on the system.

Note: Before you can use the CLI, you must create a system.

To use the CLI from a client system, complete the following steps:

- Install and set up SSH client software on each system that you plan to use to access the CLI.
- Authenticate to the system by using a password.
- Use an SSH public key if you require command line access without entering a password. Then, store the SSH public key for each SSH client on the system.

  Note: After the first SSH public key is stored, you can add SSH public keys by using either the management GUI or the CLI.

Use the CLI commands to change or create arrays, drives, enclosures, storage pools, and volumes. You can also use the CLI commands to specify encryption or security settings or work with systems.

For example, use the CLI commands to:

- Set up the system, its nodes, and the I/O groups.
- Set up and maintain canisters and enclosures.
- Analyze error logs event logs (logs).
- Set up and maintain managed disks (MDisk) and storage pools.
- Set up and maintain client public SSH keys on the system.
- Set up and maintain volumes.
- Setup logical host objects.
- Map volumes to hosts.
- Navigate from managed hosts to volumes and MDisks (and the reverse direction up the chain).
- Set up and start Copy Services functions:
  - For FlashCopy® and FlashCopy consistency groups
  - For Synchronous Metro Mirror and Metro Mirror consistency groups and relationships
  - For Asynchronous Global Mirror and Global Mirror consistency groups and relationships
  - For `active-active` consistency groups and relationships
- Setup licensing or featurization settings.

CLI commands generally give feedback whether or not the command ran. Check the audit log or event log (for configuration events, for example) after you specify a command to verify successful completion. You can also check the I/O group of the volume that you change.

Setting the clustered system time by using the CLI

You can use the command-line interface (CLI) to set the clustered system (system) time.

About this task

To set the system time:

Procedure

1. Issue the **showtimezone** CLI command to display the current time-zone settings for the system. The time zone and the associated time-zone ID are displayed.
2. Issue the **lstimezones** CLI command to list the time zones that are available on the system. A list of valid time-zone settings are displayed. Each time zone is assigned an ID. The time zone and the associated ID are indicated in the list.
3. Issue the following CLI command to set the time zone for the system.

   ```
   settimezone -timezone time_zone_setting
   ```

   where 031809142005*time_zone_setting* is the new time zone ID that you chose from the list of time zones that are available on the system.
4. Issue the following CLI command to set the time for the system:

   ```
   setsystemtime -time 031809142005
   ```

   where `031809142005` is the new time that you want to set for the system. You must use the `MMDDHHmmYYYY` format to set the time for the system.

Setting cluster date and time

You can set the date and time for a system cluster from the System Date and Time Settings panel.

Before you begin

This task assumes that you have already launched the management GUI.

About this task

You can set the System Date and time manually, or by specifying an NTP server:

Procedure

1. Click **Manage Systems** > **Set System Time** in the portfolio. The System Date and Time Settings panel is displayed.
2. To use NTP to manage the clustered system date and time, enter an Internet Protocol Version 4 (IPv4) address and click **Set NTP Server**.

   Note: If you are using a remote authentication service to authenticate users to the system, then both the system and the remote service should use the same NTP server. Consistent time settings between the two systems ensure interactive performance of the management GUI and correct assignments for user roles.
3. To set the clustered system date and time manually, continue with the following steps.
4. Type your changes into the **Date**, **Month**, **Year**, **Hours**, and **Minutes** fields and select a new time zone from the **Time Zone** list.
5. Select **Update cluster time and date**, **Update cluster time zone**, or both.
6. Click **Update** to submit the update request to the clustered system.

Viewing and updating license settings by using the CLI

You can use the command-line interface (CLI) to view and update your license settings.

About this task

Your system provides two license options: Physical Disk Licensing and Capacity Licensing. To view and update your system license settings:

Procedure

1. Issue the **lslicense** CLI command to view the current license settings for the clustered system (system).
2. Issue the **chlicense** CLI command to change the licensed settings of the system.

   Attention:

   - License settings are entered when the system is first created; do not update the settings unless you change your license.
   - To select Physical Disk Licensing, run the **chlicense** command with one or more of the **physical_disks**, **physical_flash**, and **physical_remote** parameters.
   - To select Capacity Licensing, run the **chlicense** command with one or more of the **-flash**, **-remote**, and **-virtualization** parameters. If the physical disks value is nonzero, these parameters cannot be set.

Displaying clustered system properties by using the CLI

You can use the command-line interface (CLI) to display the properties for a clustered system (system).

About this task

These actions help you display your system property information.

Procedure

Issue the **lssystem** command to display the properties for a system.
The following command is an example of the **lssystem** command you can issue:

```
lssystem -delim : build1
```

where *build1* is the name of the system.

Results

```
id:000002007A00A0FE
name:build1
location:local
partnership:
bandwidth:
total_mdisk_capacity:90.7GB
space_in_mdisk_grps:90.7GB
space_allocated_to_vdisks:14.99GB
total_free_space:75.7GB
statistics_status:on
statistics_frequency:15
required_memory:0
cluster_locale:en_US
time_zone:522 UTC
code_level:6.1.0.0 (build 47.3.1009031000)
FC_port_speed:2Gb
console_IP:9.71.46.186:443
id_alias:000002007A00A0FE
gm_link_tolerance:300
gm_inter_cluster_delay_simulation:0
gm_intra_cluster_delay_simulation:0
email_reply:
email_contact:
email_contact_primary:
email_contact_alternate:
email_contact_location:
email_state:stopped
inventory_mail_interval:0
total_vdiskcopy_capacity:15.71GB
total_used_capacity:13.78GB
total_overallocation:17
total_vdisk_capacity:11.72GB
cluster_ntp_IP_address:
cluster_isns_IP_address:
iscsi_auth_method:none
iscsi_chap_secret:
auth_service_configured:no
auth_service_enabled:no
auth_service_url:
auth_service_user_name:
auth_service_pwd_set:no
auth_service_cert_set:no
relationship_bandwidth_limit:25
gm_max_host_delay:5
tier:generic_ssd
tier_capacity:0.00MB
tier_free_capacity:0.00MB
tier:generic_hdd
tier_capacity:90.67GB
tier_free_capacity:75.34GB
email_contact2:
email_contact2_primary:
email_contact2_alternate:
total_allocated_extent_capacity:16.12GB
```

Maintaining passwords using the CLI

You can use the command-line interface (CLI) to view and change the status of the password reset feature for the system.

The superuser password can be reset to its default value of **passw0rd** by using the technician port on SAN Volume Controller 2145-DH8 nodes or the front panel on earlier models of the system. To meet varying security requirements, this functionality can be enabled or disabled by using the CLI.

Complete the following steps to view and change the status of the password reset feature:

1. Issue the **setpwdreset** CLI command to view and change the status of the password reset feature for the system.
2. Record the system superuser password because you cannot access the system without it.

The system superuser password can be reset by using a USB key. To meet varying security requirements, this functionality can be enabled or disabled by using the CLI. Complete the following steps to view and change the status of the password reset feature:

1. Issue the **setpwdreset** CLI command to view and change the status of the password reset feature for the system.
2. Record the system superuser password because you cannot access the system without it.

Using the dump commands to work with directories

The **lsdumps** command returns a list of dumps in a particular directory.

Dumps are contained in the following directory structure:

- /dumps
- /dumps/audit
- /dumps/cimom
- /dumps/elogs
- /dumps/feature
- /dumps/iostats
- /dumps/iotrace
- /dumps/mdisk
- /home/admin/update
- /dumps/drive
- /dumps/enclosure

Use the **lsdumps** command with the optional **prefix** parameter to specify a directory. If you do not specify a directory, /dumps is used as the default. Use the optional *node_id_or_name* parameter to specify the node to list the available dumps. If you do not specify a node, the available dumps on the configuration node are listed.

Use the **cpdumps** command to copy dump files from a nonconfiguration node to the configuration node. You can use this command to retrieve dumps that were saved to an older configuration node. You can retrieve files and put them on the configuration node to be copied.

Use the **cleardumps** command to delete dump directories on a specified node. You can clear specific files or groups of files based on the use of a wildcard (an asterisk, *). You can delete files on a single directory or all of the dump directories (by specifying the /dumps variable).

An audit log tracks the action commands that are issued through an SSH session or from the management GUI. To list a specified number of the most recently audited commands, issue the **catauditlog** command. To dump the contents of the audit log to a file on the current configuration node, issue the **dumpauditlog** command. This command also clears the contents of the audit log.

Dumps contained in the /dumps/cimom directory are created by the CIMOM (Common Information Model Object Manager) that runs on the clustered system (system). These files are produced during normal operations of the CIMOM.

Dumps that are contained in the /dumps/elogs directory are dumps of the contents of the error and event log at the time that the dump was taken. An error or event log dump is created by using the **dumperrlog** command. The command dumps the contents of the error or event log to the /dumps/elogs directory. If no file name prefix is supplied, the default errlog_ is used. The full default file name is errlog_*NNNNNN_YYMMDD_HHMMSS*, where *NNNNNN* is the node front panel name. If the command is used with the **-prefix** parameter, the prefix value is used instead of errlog.

Dumps that are contained in the /dumps/iostats directory are dumps of the per-node I/O statistics for disks on the system. An I/O statistics dump is created by using the **startstats** command. As part of this command, you can specify a time interval for the statistics to be written to the file; the default is 15 minutes. Every time the time interval is encountered, the I/O statistics that were collected are written to a file in the /dumps/iostats directory. The file names that are used for storing I/O statistics dumps are Nm_stats_*NNNNNN_YYMMDD_HHMMSS*, Nv_stats_*NNNNNN_YYMMDD_HHMMSS*, Nn_stats_*NNNNNN_YYMMDD_HHMMSS*, and Nd_stats_*NNNNNN_YYMMDD_HHMMSS*, where *NNNNNN* is the node name for the MDisk, volume, node, or drive.

Dumps that are contained in the /dumps/iotrace directory are dumps of I/O trace data. The type of data that is traced depends on the options that are specified by the **settrace** command. The collection of the I/O trace data is started by using the **starttrace** command. The I/O trace data collection is stopped when the **stoptrace** command is used. It is when the trace is stopped that the data is written to the file. The file name is *prefix_NNNNNN_YYMMDD_HHMMSS*, where *prefix* is the value that is entered for the **filename** parameter in the **settrace** command, and *NNNNNN* is the node name.

Dumps that are contained in the /dumps/mdisk directory are copies of flash drive MDisk internal logs. These dumps are created using the **triggerdrivedump** command. The file name is mdiskdump_*NNNNNN_MMMM_YYMMDD_HHMMSS*, where *NNNNNN* is the name of the node that contains the MDisk, and *MMMM* is the decimal ID of the MDisk.

Software update packages are contained in the /home/admin/upgrade directory. These directories exist on every node in the system.

Dumps of support data from a disk drive are contained in the /dumps/drive directory. This data can help to identify problems with the drive, and does not contain any data that applications might have written to the drive.

Dumps from an enclosure or enclosures are contained in the /dumps/enclosure directory.

Dumps that are contained in the /dumps directory result from application abends. Such dumps are written to the /dumps directory. The default file names are dump.*NNNNNN.YYMMDD.HHMMSS*, where *NNNNNN* is the node front panel name. In addition to the dump file, there might be some trace files written to this directory that are named *NNNNNN*.trc.

Because files can only be copied from the current configuration node (by using secure copy), you can issue the **cpdumps** command to copy the files from a nonconfiguration node to the current configuration node.

Re-adding a repaired node to a clustered system by using the CLI

You can use the command-line interface (CLI) to re-add a failed node back into a clustered system after it was repaired.

Before you begin

Before you add a node to a clustered system, you must make sure that the switchd\ zoning is configured such that the node that is being added is in the same zone as all other nodes in the clustered system. If you are replacing a node and the switch is zoned by worldwide port name (WWPN) rather than by switch port, make sure that the switch is configured such that the node that is being added is in the same VSAN/zone.

Attention:

1. If you are re-adding a node to the SAN, ensure that you are adding the node to the same I/O group from which it was removed. Failure to select the correct I/O group can result in data corruption. You must use the information that was recorded when the node was originally added to the clustered system. If you do not have access to this information, call the IBM Support Center to add the node back into the clustered system without corrupting the data.
2. The LUNs that are presented to the ports on the new node must be the same as the LUNs that are presented to the nodes that currently exist in the clustered system. You must ensure that the LUNs are the same before you add the new node to the clustered system.
3. LUN masking for each LUN must be identical on all nodes in a clustered system. You must ensure that the LUN masking for each LUN is identical before you add the new node to the clustered system.
4. You must ensure that the model type of the new node is supported by the SAN Volume Controller software level that is installed on the clustered system. If the model type is not supported by the SAN Volume Controller software level, update the clustered system to a software level that supports the model type of the new node. See the following website for the latest supported software levels:

   www.ibm.com/support

About this task

Special procedures when you add a node to a clustered system

Applications on the host systems direct I/O operations to file systems or logical volumes that are mapped by the operating system to virtual paths (vpaths), which are pseudo disk objects that are supported by the Subsystem Device Driver (SDD). SDD maintains an association between a vpath and a SAN Volume Controller volume. This association uses an identifier (UID) which is unique to the volume and is never reused. The UID permits SDD to directly associate vpaths with volumes.

SDD operates within a protocol stack that contains disk and Fibre Channel device drivers that are used to communicate with the SAN Volume Controller using the SCSI protocol over Fibre Channel as defined by the ANSI FCS standard. The addressing scheme that is provided by these SCSI and Fibre Channel device drivers uses a combination of a SCSI logical unit number (LUN) and the worldwide node name (WWNN) for the Fibre Channel node and ports.

If an error occurs, the error recovery procedures (ERPs) operate at various tiers in the protocol stack. Some of these ERPs cause I/O to be redriven by using the same WWNN and LUN numbers that were previously used.

SDD does not check the association of the volume with the vpath on every I/O operation that it performs.

Before you add a node to the clustered system, you must check to see if any of the following conditions are true:

- The clustered system has more than one I/O group.
- The node that is being added to the clustered system uses physical node hardware or a slot that has previously been used for a node in the clustered system.
- The node that is being added to the clustered system uses physical node hardware or a slot that has previously been used for a node in another clustered system and both clustered systems have visibility to the same hosts and back-end storage.

If any of the previous conditions are true, the following special procedures apply:

- The node must be added to the same I/O group that it was previously in. You can use the command-line interface (CLI) command **lsnode** or the management GUI to determine the WWN of the clustered system nodes.

- Before you add the node back into the clustered system, you must shut down all of the hosts using the clustered system. The node must then be added before the hosts are rebooted. If the I/O group information is unavailable or it is inconvenient to shut down and reboot all of the hosts by using the clustered system, then do the following:
  - On all of the hosts that are connected to the clustered system, unconfigure the Fibre Channel adapter device driver, the disk device driver, and multipathing driver before you add the node to the clustered system.
  - Add the node to the clustered system, and then reconfigure the Fibre Channel adapter device driver, the disk device driver, and multipathing driver.

Scenarios where the special procedures can apply

The following two scenarios describe situations where the special procedures can apply:

- Four nodes of an eight-node clustered system have been lost because of the failure of a pair of 2145 UPS or four 2145 UPS-1U. In this case, the four nodes must be added back into the clustered system by using the CLI command **addnode** or the management GUI.

  Note: You do not need to run the **addnode** command on a node with a partner that is already in a clustered system; the clustered system automatically detects an online candidate.

  Note: The **addnode** command is a SAN Volume Controller command. For Storwize V7000, use the **addcontrolenclosure**command.
- A user decides to delete four nodes from the clustered system and add them back into the clustered system using the CLI command **addnode** or the management GUI.

  Note: The **addnode** command is a SAN Volume Controller command. For Storwize V7000, use the **addcontrolenclosure** command.

For 5.1.0 nodes, the SAN Volume Controller automatically re-adds nodes that failed back to the clustered system. If the clustered system reports an error for a node missing (error code 1195) and that node has been repaired and restarted, the clustered system automatically re-adds the node back into the clustered system. This process can take up to 20 minutes, so you can manually re-add the node by completing the following steps:

Procedure

1. Issue the **lsnode** CLI command to list the nodes that are currently part of the clustered system and determine the I/O group for which to add the node.

   The following is an example of the output that is displayed:

   ```
   lsnode -delim :

   id:name:UPS_serial_number:WWNN:status:IO_group_id:IO_group_name
   :config_node:UPS_unique_id:hardware:iscsi_name:iscsi_alias
   :panel_name:enclosure_id:canister_id:enclosure_serial_number
   1:node1::50050868010050B2:online:0:io_grp0:yes::100:iqn.1986-03.com.ibm
   :2145.cluster0.node1::02-1:2:1:123ABCG
   2:node2::50050869010050B2:online:0:io_grp0:no::100:iqn.1986-03.com.ibm
   :2145.cluster0.node2::02-2:2:2:123ABDG
   ```

   Storwize V7000 example:

```
lsnode -delim :

id:name:UPS_serial_number:WWNN:status:IO_group_id:IO_group_name
:config_node:UPS_unique_id:hardware:iscsi_name:iscsi_alias
:panel_name:enclosure_id:canister_id:enclosure_serial_number
1:node1::50050868010050B2:online:0:io_grp0:yes::100:iqn.1986-03.com.ibm
:2145.cluster0.node1::02-1:2:1:123ABCG
2:node2::50050869010050B2:online:0:io_grp0:no::100:iqn.1986-03.com.ibm
:2145.cluster0.node2::02-2:2:2:123ABDG
```

2. Issue the **lsnodecandidate** CLI command to list nodes that are not assigned to a clustered system and to verify that a second node is added to an I/O group.

   Note: The **`lsnodecandidate`** command is a SAN Volume Controller command. For Storwize V7000, use the **`lscontrolenclosurecandidate`** command.

   The following is an example of the output that is displayed:

   ```
   lsnodecandidate -delim :

   id:panel_name:UPS_serial_number:UPS_unique_id:hardware
   5005076801000001:000341:10L3ASH:202381001C0D18D8:8A4
   5005076801000009:000237:10L3ANF:202381001C0D1796:8A4
   50050768010000F4:001245:10L3ANF:202381001C0D1796:8A4
   ....
   ```

3. Issue the **addnode** CLI command to add a node to the clustered system.

   Note: The **`addnode`** command is a SAN Volume Controller command. For Storwize V7000, use the **`addcontrolenclosure`** command.

   Important: Each node in an I/O group must be attached to a different uninterruptible power supply.

   The following is an example of the CLI command you can issue to add a node to the clustered system by using the panel name parameter:

   ```
   addnode -panelname 000237
    -iogrp io_grp0
   ```

   Where *000237* is the panel name of the node, *io_grp0* is the name of the I/O group that you are adding the node to.

   The following is an example of the CLI command you can issue to add a node to the clustered system by using the WWNN parameter:

   ```
   addnode -wwnodename 5005076801000001
    -iogrp io_grp1
   ```

   Where *5005076801000001* is the WWNN of the node, *io_grp1* is the name of the I/O group that you are adding the node to.

4. Issue the **`lsnode`** CLI command to verify the final configuration.

   The following example shows output that is displayed:

   ```
   lsnode -delim :

   id:name:UPS_serial_number:WWNN:status:IO_group_id:IO_group_name:config_node:UPS_unique_id:
   hardware:iscsi_name:iscsi_alias
   1:node1:10L3ASH:0000000000000000:offline:0:io_grp0:no:1000000000003206:
   8A4:iqn.1986-03.com.ibm:2145.ndihill.node1:
   ```

   Record the following information for the new node:

   - Node name
   - Node serial number
   - WWNN

- IQNs (if using hosts attached by using iSCSI connections)
- All WWPNs
- I/O group that contains the node

Note: If this command is issued quickly after you add nodes to the clustered system, the status of the nodes might be adding. The status is shown as adding if the process of adding the nodes to the clustered system is still in progress. You do not have to wait for the status of all the nodes to be online before you continue with the configuration process.

Results

The nodes are added to the clustered system.

Displaying node properties by using the CLI

You can use the command-line interface (CLI) to display node properties.

About this task

To display the node properties:

Procedure

1. Use the **lsnode** CLI command to display a concise list of nodes in the clustered system.

   Issue this CLI command to list the system nodes:

   ```
   lsnode -delim :
   ```
2. Issue the **lsnode** CLI command and specify the node ID or name of the node that you want to receive detailed output.

   The following example is a CLI command that you can use to list detailed output for a node in the system:

   ```
   lsnode -delim : group1node1
   ```

   Where *group1node1* is the name of the node for which you want to view detailed output.

Discovering MDisks using the CLI

You can use the command-line interface (CLI) to discover managed disks (MDisks).

About this task

The clustered system (system) automatically discovers the back-end controller and integrates the controller to determine the storage that is presented to the system nodes when back-end controllers are:

- Added to the Fibre Channel
- Included in the same switch zone as a system

The Small Computer System Interface (SCSI) logical units (LUs) that are presented by the back-end controller are displayed as unmanaged MDisks. However, if the configuration of the back-end controller is modified after this has occurred, the system might be unaware of these configuration changes. You can request that the system rescan the Fibre Channel SAN to update the list of unmanaged MDisks.

Note: The automatic discovery completed by the system does not write anything to an unmanaged MDisk. You must instruct the system to add an MDisk to a storage pool or use an MDisk to create an image mode volume.

Discover (and then view) a list of MDisks:

Procedure

1. Issue the **detectmdisk** CLI command to manually scan the Fibre Channel network. The scan discovers any new MDisks that might have been added to the system and can help rebalance MDisk access across the available controller device ports.

   Notes:

   a. Only issue the **detectmdisk** command when you are sure that all of the disk controller ports are working and correctly configured in the controller and the SAN zoning. Failure to do this can result in errors that are not reported.

   b. Although it might appear that the **detectmdisk** command has completed, extra time might be required for it to run. The **detectmdisk** is asynchronous and returns a prompt while the command continues to run in the background. You can use the **lsdiscoverystatus** command to view the discovery status.

2. When the detection is complete, issue the **lsmdiskcandidate** CLI command to show the unmanaged MDisks. These MDisks have not been assigned to a storage pool.
3. Issue the **lsmdisk** CLI command to view all of the MDisks.

Results

You have now seen that the back-end controllers and switches have been set up correctly and that the system recognizes the storage that is presented by the back-end controller.

Example

This example describes a scenario where a single back-end controller is presenting eight SCSI LUs to the system:

1. Issue `detectmdisk`.
2. Issue `lsmdiskcandidate`.

   This output is displayed:

```
id
0
1
2
3
4
5
6
7
```

3. Issue `lsmdisk -delim :`

   This output is displayed:

```
lsmdisk -delim :
id:name:status:mode:mdisk_grp_id:mdisk_grp_name:capacity:ctrl_LUN_#:controller_name:UID:tier
0:mdisk0:online:unmanaged:::68.4GB:0000000000000000:controller0:
20000004cf2422aa000000000000000000000000000000000000000000000000:
1:mdisk1:online:unmanaged:::68.4GB:0000000000000000:controller1:
20000004cf1fd19d000000000000000000000000000000000000000000000000:
2:mdisk2:online:unmanaged:::68.4GB:0000000000000000:controller2:
20000004cf242531000000000000000000000000000000000000000000000000:
```

Creating storage pools using the CLI

You can use the command-line interface (CLI) to create a storage pool.

Before you begin

Attention: If you add an MDisk to a storage pool as an MDisk, any data on the MDisk is lost. If you want to keep the data on an MDisk (for example, because you want to import storage that was previously not managed by the system), you must create image mode volumes instead.

Assume that the system has been set up and that a back-end controller has been configured to present new storage to the system.

If you are using a flash drive managed disk on your system, ensure that you are familiar with the flash drive configuration rules.

If you intend to keep the volume allocation within one storage system, ensure that all MDisks in the storage pool are presented by the same storage system.

Ensure that all MDisks that are allocated to a single storage pool are of the same RAID type. If the storage pool has more than one tier of storage, ensure that all MDisks in the same tier are of the same RAID type. When using Easy Tier®, all of the MDisks in a storage pool in the same tier must be similar and have similar performance characteristics. If you do not use Easy Tier, the storage pool must contain only one tier of storage, and all of the MDisks in the storage pool must be similar and have similar performance characteristics.

As you plan how many pools to create, consider the following factors:

- A volume can only be created using the storage from one storage pool. Therefore, if you create small (storage pools), you might lose the benefits that are provided by virtualization, namely more efficient management of free space and a more evenly distributed workload for better performance.
- If any MDisk in an storage pool goes offline, all the (volumes) in the storage pool go offline. Therefore you might want to consider using different storage pools for different back-end controllers or for different applications.
- If you anticipate regularly adding and removing back-end controllers or storage, this task is made simpler by grouping all the MDisks that are presented by a back-end controller into one storage pool.
- All the MDisks in a storage pool must have similar levels of performance or reliability, or both. If a storage pool contains MDisks with different levels of performance, the performance of the (volumes) in this group is limited by the performance of the slowest MDisk. If a storage pool contains MDisks with different levels of reliability, the reliability of the (volumes) in this group is that of the least reliable MDisk in the group.

Note: When you create a pool with a new flash drive, the new flash drive is automatically formatted and set to a block size of 512 bytes.

About this task

Even with the best planning, circumstances can change and you must reconfigure your (storage pools) after they have been created. The data migration facilities that are provided by the system enable you to move data without disrupting I/O.

Choosing a storage pool extent size

As you plan the extent size of each new pool, consider the following factors:

- You must specify the extent size when you create a new storage pool.
- You cannot change the extent size later; it must remain constant throughout the lifetime of the storage pool.
- Storage pools can have different extent sizes; however, this places restrictions on the use of data migration.

- The extent size affects the maximum size of a volume in the storage pool. A larger extent size increases the total amount of storage that the system can manage, and a smaller extent size allows more fine-grained control of storage allocation.

Table 6 compares the maximum volume capacity for each extent size. The maximum is different for thin-provisioned volumes. Because the system allocates a whole number of extents to each volume that is created, using a larger extent size might increase the amount of storage that is wasted at the end of each volume. Larger extent sizes also reduces the ability of the system to distribute sequential I/O workloads across many MDisks and therefore can reduce the performance benefits of virtualization.

Table 6. Maximum volume capacity by extent size

Extent size (MB)	Maximum volume capacity in GB (not thin-provisioned volumes)	Maximum volume capacity in GB (thin-provisioned volumes)
16	2048 (2 TB)	2000
32	4096 (4 TB)	4000
64	8192 (8 TB)	8000
128	16,384 (16 TB)	16,000
256	32,768 (32 TB)	32,000
512	65,536 (64 TB)	65,000
1024	131,072 (128 TB)	130,000
2048	262,144 (256 TB)	260,000
4096	262,144 (256 TB)	262,144
8192	262,144 (256 TB)	262,144

Important: You can specify different extent sizes for different storage pools; however, you cannot migrate (volumes) between storage pools with different extent sizes. If possible, create all your storage pools with the same extent size.

Use the following steps to create a storage pool:

Procedure

Issue the **mkmdiskgrp** CLI command to create a storage pool.
This is an example of the CLI command you can issue to create a storage pool:

```
mkmdiskgrp -name maindiskgroup -ext 32
 -mdisk mdsk0:mdsk1:mdsk2:mdsk3
```

where *maindiskgroup* is the name of the storage pool that you want to create, *32* MB is the size of the extent you want to use, and *mdsk0, mdsk1, mdsk2, mdsk3* are the names of the four MDisks that you want to add to the group.

Results

You created and added MDisks to a storage pool.

Example

The following example provides a scenario where you want to create a storage pool, but you do not have any MDisks available to add to the group. You plan to add the MDisks at a later time. You use the **mkmdiskgrp** CLI command to create the storage pool *bkpmdiskgroup* and later used the **addmdisk** CLI command to add *mdsk4, mdsk5, mdsk6, mdsk7* to the storage pool.

1. Issue `mkmdiskgrp -name bkpmdiskgroup -ext 32`

where *bkpmdiskgroup* is the name of the storage pool that you want to create and *32* MB is the size of the extent that you want to use.

2. You find four MDisks that you want to add to the storage pool.
3. Issue `addmdisk -mdisk mdsk4:mdsk5:mdsk6:mdsk7 bkpdiskgroup`

   where *mdsk4, mdsk5, mdsk6, mdsk7* are the names of the MDisks that you want to add to the storage pool and *bkpdiskgroup* is the name of the storage pool for which you want to add MDisks.

Adding MDisks to storage pools using the CLI

You can use the command-line interface (CLI) to add managed disks (MDisks) to storage pools.

Before you begin

The MDisks must be in unmanaged mode. Disks that already belong to a storage pool cannot be added to another storage pool until they have been deleted from their current storage pool. An MDisk can be deleted from a storage pool under these circumstances:

- If the MDisk does not contain any extents in use by a volume
- If you can first migrate the extents in use onto other free extents within the group

About this task

Important: Do not add an MDisk using this procedure if you are mapping the MDisk to an image mode volume. Adding an MDisk to a storage pool enables the system to write new data to the MDisk; therefore, any existing data on the MDisk is lost. If you want to create an image mode volume, use the **mkvdisk** command instead of **addmdisk**.

If you are using a flash drive managed disk on your system, ensure that you are familiar with the flash drive configuration rules.

The system performs tests on the MDisks in the list before the MDisks are allowed to become part of a storage pool when:

- Adding MDisks to a storage pool using the **addmdisk** command
- Creating a storage pool using the **mkmdiskgrp -mdisk** command

These tests include checks of the MDisk identity, capacity, status and the ability to perform both read and write operations. If these tests fail or exceed the time allowed, the MDisks are not added to the group. However, with the **mkmdiskgrp -mdisk** command, the storage pool is still created even if the tests fail, but it does not contain any MDisks. If tests fail, confirm that the MDisks are in the correct state and that they have been correctly discovered.

These events contribute to an MDisk test failure:

- The MDisk is not visible to all system nodes in the clustered system.
- The MDisk identity has changed from a previous discovery operation.
- The MDisk cannot perform read or write operations.
- The status of the MDisk can be either degraded paths, degraded ports, excluded, or offline.
- The MDisk does not exist.

These events contribute to an MDisk test timeout:

- The disk controller system on which the MDisk resides is failing.
- A SAN fabric or cable fault condition exists that is preventing reliable communication with the MDisk.

Note: The first time that you add a new flash drive to a pool, the flash drive is automatically formatted and set to a block size of 512 bytes.

Procedure

To add MDisks to storage pools, complete the following steps.

1. Issue the **lsmdiskgrp** CLI command to list the existing storage pools.

   This example is a CLI command that you can issue to list the existing storage pools:

   ```
   lsmdiskgrp -delim :
   ```

   This is an example of the output that is displayed:

   ```
   id:name:status:mdisk_count:vdisk_count:
   capacity:extent_size:free_capacity:virtual_capacity:
   used_capacity:real_capacity:overallocation:warning
   0:mdiskgrp0:online:3:4:33.3GB:16:32.8GB:64.00MB:64.00MB:64.00MB:0:0
   1:mdiskgrp1:online:2:1:26.5GB:16:26.2GB:16.00MB:16.00MB:16.00MB:0:0
   2:mdiskgrp2:online:2:0:33.4GB:16:33.4GB:0.00MB:0.00MB:0.00MB:0:0
   ```

2. Issue the **addmdisk** CLI command to add MDisks to the storage pool.

   This is an example of the CLI command you can issue to add MDisks to a storage pool:

   ```
   svctask addmdisk -mdisk mdisk4:mdisk5:mdisk6:mdisk7 bkpmdiskgroup
   ```

   Where *mdisk4:mdisk5:mdisk6:mdisk7* are the names of the MDisks that you want to add to the storage pool and *bkpmdiskgroup* is the name of the storage pool for which you want to add the MDisks.

Setting a quorum disk using the CLI

You can set an external managed disk (MDisk) as a quorum disk by using the command-line interface (CLI).

Note: Quorum functionality is not supported for internal drives on nodes.

To set an MDisk as a quorum disk, use the **chquorum** command. **Storwize V7000:** To set an external MDisk as a quorum disk, use the **chquorum** command.

When setting an MDisk as a quorum disk, keep the following recommendations in mind:

- When possible, distribute the quorum candidate disks so that each MDisk is provided by a different storage system. For a list of storage systems that support quorum disks, search for `supported hardware list` at the following website:

  www.ibm.com/support
- Before you set the quorum disk with the **chquorum** command, use the **lsmdisk** or **lsdrive** command to ensure that the MDisk you want is online. If you want to set a drive as quorum, use **lsdrive** to make sure it is online. If you want to set an MDisk as quorum, use **lsmdisk** to make sure it is online.

Quorum disk configuration describes how quorum disks are used by the system, and how they are selected. The system automatically assigns quorum disks. Do not override the quorum disk assignment if you have a system without external MDisks. For a system with more than one control enclosure and with external MDisks, distribute the quorum candidate disks (when possible) so that each MDisk is provided by a different storage system. For a list of storage systems that support quorum disks, search for `supported hardware list` at the following website:

www.ibm.com/support

Modifying the amount of available memory for Copy Services, Volume Mirroring, and RAID arrays by using the CLI

You can use the command-line interface (CLI) to modify the amount of memory that is available for RAID arrays, the volume mirroring feature, and the FlashCopy, Metro Mirror, Global Mirror, or HyperSwap® active-active Copy Services features.

About this task

Copy Services features and RAID require that small amounts of volume cache be converted from cache memory into bitmap memory to allow the functions to operate. If you do not have enough bitmap space allocated when you try to use one of the functions, you will not be able to complete the configuration.

The total memory that can be dedicated to these functions is not defined by the physical memory in the system. The memory is constrained by the software functions that use the memory.

In planning the installation for a system, consider the future requirements for the advanced functions.

The following tables describe the amount of bitmap space necessary to configure the various Copy Services functions and RAID:

This table provides an example of the amount of memory that is required for remote mirroring functions, FlashCopy functions, and volume mirroring.

Table 7. Examples of memory required

Function	**Grain size**	**1 MiB of memory provides the following volume capacity for the specified I/O group**
Remote copy	256 KiB	2 TiB of total Metro Mirror, Global Mirror, or HyperSwap volume capacity
FlashCopy	256 KiB	2 TiB of total FlashCopy source volume capacity
FlashCopy	64 KiB	512 GiB of total FlashCopy source volume capacity
Incremental FlashCopy	256 KiB	1 TiB of total incremental FlashCopy source volume capacity
Incremental FlashCopy	64 KiB	256 GiB of total incremental FlashCopy source volume capacity
Volume mirroring	256 KiB	2 TiB of mirrored volume capacity

Notes:

1. For multiple FlashCopy targets, you must consider the number of mappings. For example, for a mapping with a grain size of 256 KiB, 8 KiB of memory allows one mapping between a 16 GiB source volume and a 16 GiB target volume. Alternatively, for a mapping with a 256 KiB grain size, 8 KiB of memory allows two mappings between one 8 GiB source volume and two 8 GiB target volumes.
2. When creating a FlashCopy mapping, if you specify an I/O group other than the I/O group of the source volume, the memory accounting goes toward the specified I/O group, not toward the I/O group of the source volume.
3. For volume mirroring, the full 512 MiB of memory space enables 1 PiB of total volume mirroring capacity.
4. When creating new FlashCopy relationships or mirrored volumes, additional bitmap space is allocated automatically by the system if required.

Table 8 provides an example of RAID level comparisons with their bitmap memory cost, where *MS* is the size of the member drives and *MC* is the number of member drives.

Table 8. RAID level comparisons

Level	Member count	Approximate capacity	Redundancy	Approximate bitmap memory cost
RAID-0	1-8	MC * MS	None	(1 MB per 2 TB of MS) * MC
RAID-1	2	MS	1	(1 MB per 2 TB of MS) * (MC/2)
RAID-5	3-16	(MC-1) * MS	1	1 MB per 2 TB of MS with a strip size of 256 KB; double with strip size of 128 KB.
RAID-6	5-16	less than (MC-2 * MS)	2	
RAID-10	2-16 (evens)	MC/2 * MS	1	(1 MB per 2 TB of MS) * (MC/2)
Note: There is a margin of error on the approximate bitmap memory cost of approximately 15%. For example, the cost for a 256 KB strip size for RAID-5 is ~1.15 MB for the first 2 TB of MS.				

Before you specify the configuration changes, consider the following factors:

- For FlashCopy mappings, only one I/O group consumes bitmap space. By default, the I/O group of the source volume is used.
- For Metro Mirror, Global Mirror, and HyperSwap active-active relationships, two bitmaps exist. For Metro Mirror or Global Mirror relationships, one is used for the master system and one is used for the auxiliary system because the direction of the relationship can be reversed. For active-active relationships, which are configured automatically when HyperSwap volumes are created, one bitmap is used for the volume copy on each site because the direction of these relationships can be reversed.
- When you create a reverse mapping; for example, to run a restore operation from a snapshot to its source volume; a bitmap is also created for this reverse mapping.
- When you configure change volumes for use with Global Mirror or Metro Mirror, two internal FlashCopy mappings are created for each change volume.
- The smallest possible bitmap is 4 KiB; therefore, a 512 byte volume requires 4 KiB of bitmap space.

On existing systems, also consider these factors:

- When you create FlashCopy mappings and mirrored volumes, HyperSwap volumes, or formatted, fully allocated volumes, the system attempts to automatically increase the available bitmap space. You do not need to manually increase this space.
- Metro Mirror and Global Mirror relationships do not automatically increase the available bitmap space. You might need to use the **chiogrp** command or the management GUI to manually increase the space in one or both of the master and auxiliary systems.

To modify and verify the amount of memory that is available, complete the following steps:

Procedure

1. Issue the following command to modify the amount of memory that is available for Volume Mirroring or a Copy Service feature:
   `chiogrp -feature` *flash* | *remote* | *mirror* `-size` *memory_size io_group_id* | *io_group_name*

   where *flash* | *remote* | *mirror* is the feature that you want to modify, *memory_size* is the amount of memory that you want to be available, and *io_group_id* | *io_group_name* is the ID or name of the I/O group for which you want to modify the amount of available memory.
2. Issue the following command to verify that the amount of memory has been modified:
   `lsiogrp` *object_id* | *object_name*

where *object_id* | *object_name* is the ID or name of the I/O group for which you have modified the amount of available memory.

The following information is an example of the output that is displayed.

```
id 0
name io_grp0
node_count 2
vdisk_count 40
host_count 1
flash_copy_total_memory 5.0MB
flash_copy_free_memory 5.0MB
remote_copy_total_memory 20.0MB
remote_copy_free_memory 20.0MB
mirroring_total_memory 20.0MB
mirroring_free_memory 20.0MB
raid_total_memory 40.0MB
raid_free_memory 0.1MB
maintenance no
compression_active no
accessible_vdisk_count 40
compression_supported yes
max_enclosures 21
encryption_supported yes
```

Creating volumes using the CLI

You can use the command-line interface (CLI) to create a volume. You can create volumes that are not high availability volumes or you can create high availability volumes.

Before you begin

If the volume that you are creating maps to a flash drive, the data that is stored on the volume is not protected against Flash drive failures or node failures. To avoid data loss, add a volume copy that maps to a Flash drive on another node.

This task assumes that the clustered system (system) has been set up and that you have created storage pools. You can establish an empty storage pool to hold the MDisks that are used for image mode volumes.

About this task

Note: If you want to keep the data on an MDisk, create image mode (volumes). This task describes how to create a volume with striped virtualization.

Use the **mkvdisk** command to create sequential, striped, or image mode volumes that are not high availability volumes. Use the **mkvolume** command to create high availability volumes (or volumes that are not high availability). Use the **mkimagevolume** command to create an image mode volume by importing (preserving) data on a managed disk from another storage system.

Procedure

To create volumes, complete these steps.

1. Issue the **lsmdiskgrp** CLI command to list the available storage pools and the amount of free storage in each group.

   Issue this CLI command to list storage pools:

   ```
   lsmdiskgrp -delim :
   ```

   This output is displayed:

```
id:name:status:mdisk_count:vdisk_count:capacity:extent_size:free_capacity:virtual_capacity:
used_capacity:real_capacity:overallocation:warning:easy_tier:easy_tier_status
0:mdiskgrp0:degraded:4:0:34.2GB:16:34.2GB:0:0:0:0:0:auto:inactive
1:mdiskgrp1:online:4:6:200GB:16:100GB:400GB:75GB:100GB:200:80:on:active
```

2. Decide which storage pool you want to provide the storage for the volume.
3. Issue the **lsiogrp** CLI command to show the I/O groups and the number of volumes assigned to each I/O group.

   Note: It is normal for systems with more than one I/O group to have **`mkvdisk`** that have volumes in different I/O groups. You can use FlashCopy to make copies of volumes regardless of whether the source and target volume are in the same I/O group.

   Similarly, if you plan to use intra-system Metro Mirror or Global Mirror, both the master and auxiliary volume can be in the same I/O group or different I/O groups.

   Issue this CLI command to list I/O groups:

   ```
   lsiogrp -delim :
   ```

   This output is displayed:

   ```
   id:name:node_count:vdisk_count:host_count
   0:io_grp0:2:0:2
   1:io_grp1:2:0:1
   2:io_grp2:0:0:0
   3:io_grp3:0:0:0
   4:recovery_io_grp:0:0:0
   ```

4. Decide which I/O group you want to assign the volume to. This determines which system nodes in the system process the I/O requests from the host systems. If you have more than one I/O group, make sure you distribute the volumes between the I/O groups so that the I/O workload is shared evenly between all system nodes.
5. Issue the **mkvdisk** CLI command to create a volume (that is not a high availability volume) that uses striped virtualization. Use the **`mkvolume`** command to create high availability volumes.

   The rate at which the volume copies resynchronize after loss of synchronization can be specified by using the **`-syncrate`** parameter. Table 9 defines the rates. These settings also affect the initial rate of formatting.

Table 9. Volume copy resynchronization rates

Syncrate value	**Data copied per second**
1-10	128 KB
11-20	256 KB
21-30	512 KB
31-40	1 MB
41-50	2 MB
51-60	4 MB
61-70	8 MB
71-80	16 MB
81-90	32 MB
91-100	64 MB

The default setting is 50. The synchronization rate must be set such that the volume copies resynchronize quickly after loss of synchronization.

Issue this CLI command to create a volume with two copies using the I/O group and storage pool name and specifying the synchronization rate:

```
mkvdisk -iogrp io_grp1 -mdiskgrp grpa:grpb -size500 -vtype striped
-copies 2 –syncrate 90
```

where *io_grp1* is the name of the I/O group that you want the volume to use, *grpa* is the name of the storage pool for the primary copy of the volume and *grpb* is the name of the storage pool for the second copy of the volume, and 2 is the number of volume copies and the synchronization rate is 90 which is equivalent to 32MB per second.

Issue this CLI command to create a volume using the I/O group ID and storage pool ID:

```
mkvdisk -name mainvdisk1 -iogrp 0
 -mdiskgrp 0 -vtype striped -size 256 -unit gb
```

where *mainvdisk1* is the name that you want to call the volume, *0* is the ID of the I/O group that want the volume to use, *0* is the ID of the storage pool that you want the volume to use, and *256* is the capacity of the volume.

Issue this CLI command to create a thin-provisioned volume using the I/O group and storage pool name:

```
mkvdisk  -iogrp io_grp1 -mdiskgrp bkpmdiskgroup -vtype striped
-size 10 unit gb -rsize 20% -autoexpand -grainsize 32
```

where *io_grp1* is the name of the I/O group that you want the volume to use and *20%* is how much real storage to allocate to the volume, as a proportion of its virtual size. In this example, the size is 10 GB so that 2 GB will be allocated.

Issue this CLI command to create a volume with two copies using the I/O group and storage pool name:

```
mkvdisk -iogrp io_grp1 -mdiskgrp grpa:grpb
-size 500 -vtype striped -copies 2
```

where *io_grp1* is the name of the I/O group that you want the volume to use, *grpa* is the name of the storage pool for the primary copy of the volume and *grpb* is the name of the storage pool for the second copy of the volume, and 2 is the number of volume copies.

Issue this CLI command to create a striped high availability volume:

```
mkvolume -pool 0:1 -size 1000
```

This creates a volume in storage pool `0` with a capacity of `1000` MBs.

Issue this CLI command to create an image mode volume:

```
mkimagevolume -mdisk 7 -pool 1 -thin -size 25 -unit gb
```

This imports a thin-provisioned image mode volume with a virtual capacity of `25` GB in storage pool `1` using MDisk `7`.

Note: If you want to create two volume copies of different types, create the first copy using the **mkvdisk** command and then add the second copy using the **addvdiskcopy** command.To create a high availability volume, use the **mkvolume** command. To convert a basic volume to a high availability volume use the **addvolumecopy** command.

6. Issue the **lsvdisk** CLI command to list all the volumes that have been created.

Adding a copy to a volume

You can use the management GUI or command-line interface (CLI) to add a mirrored copy to a volume. Each volume can have a maximum of two copies.

Before you begin

The system supports mirrored copies for both standard topology, which consists of a single site, and HyperSwap and stretched system topologies, which consist of multiple sites. Both HyperSwap and stretched system topologies are used for high availability configurations. However, in HyperSwap topologies, separate I/O groups are at each site. For stretched system, individual I/O groups are split between sites with each node in the I/O group at separate sites. If you are adding a mirrored copy to a volume used in a standard topology, use the **addvdiskcopy** command or the management GUI. For HyperSwap and stretched volumes, use the **addvolumecopy**. In the management GUI, select **Volumes**, right-click the volume, and select **Add Volume Copies**.

Deleting a copy from a volume

You can use the management GUI or command-line interface (CLI) to delete a copy from a volume.

Before you begin

The system supports mirrored copies for both standard topology, which consists of a single site, and HyperSwap and stretched system topologies, which consist of multiple sites. Both HyperSwap and stretched system topologies are used for high availability configurations. However, in HyperSwap topologies, separate I/O groups are at each site. For stretched system, individual I/O groups are split between sites with each node in the I/O group at separate sites.The system supports mirrored copies for both standard topology, which consists of a single site, and HyperSwap and stretched system topologies, which consist of multiple sites. Both HyperSwap and stretched system topologies are used for high availability configurations. However, in HyperSwap topologies, separate I/O groups are at each site. For stretched system, individual I/O groups are split between sites with each node in the I/O group at separate sites. If you are deleting a copy to a volume used in a single system, use the **rmvdiskcopy** command or the management GUI. For HyperSwap volumes, use the **rmvolumecopy**. In the management GUI, select **Volumes**, right-click the volume copy, and select **Delete this Copy**.

Configuring host objects

You can use the management GUI or command-line interface (CLI) to create host objects.

Before you begin

If you are configuring a host object on a Fibre Channel attached host, ensure that you have completed all zone and switch configuration. Also test the configuration to ensure that zoning was created correctly.

If you are configuring a host object on the clustered system (system) that uses iSCSI connections, ensure that you have completed the necessary host-system configurations and have configured the system for iSCSI connections.

At least one WWPN or iSCSI name must be specified.

To create a host object in the management GUI, select **Hosts** > **Hosts** > **Add Hosts**.

About this task

To create host objects with the command-line interface, use the following steps:

Procedure

1. Issue the **mkhost** CLI command to create a logical host object for a Fibre Channel attached host. Assign your worldwide port name (WWPN) for the host bus adapters (HBAs) in the hosts.

This is an example of the CLI command that you can issue to create a Fibre Channel attached host:

```
mkhost -name new_name -fcwwpn wwpn_list
```

where *new_name* is the name of the host and *wwpn_list* is the WWPN of the HBA.

2. To create an iSCSI-attached host, issue the following CLI command:

```
mkhost -iscsiname iscsi_name_list
```

where *iscsi_name_list* specifies one or more iSCSI qualified names (IQNs) of this host. Up to 16 names can be specified, provided that the command-line limit is not reached. Each name should comply with the iSCSI standard, RFD 3720.

3. To add ports to a Fibre Channel attached host, issue the **addhostport** CLI command.

For example, issue the following CLI command:

```
addhostport -hbawwpn wwpn_list new_name
```

This command adds another HBA WWPN *wwpn_list* to the host that was created in step 1 on page 31.

4. To add ports to an iSCSI-attached host, issue the **addhostport** CLI command.

For example, issue the following CLI command:

```
addhostport -iscsiname iscsi_name_list new_name
```

where *iscsi_name_list* specifies the comma-separated list of IQNs to add to the host. This command adds an IQN to the host that was created in step 2.

5. To set up Challenge Handshake Authentication Protocol (CHAP) to authenticate iSSCI-attached hosts, issue the **chhost** CLI command. The system supports both one-way and two-way CHAP authentication. In one-way CHAP authentication, the system authenticates to the host and with two-way chap authentication, both the host and the system authenticate to each other. You can specify the one-way chap secret and the user name for that host object by using the **chhost** command that will be used in one-way chap authentication. For example, issue the following CLI command:

   ```
   chhost -chapsecret chap_secret –iscsiusername  username
   ```

   where *chap_secret* is the CHAP secret that is used to authenticate the host for iSCSI I/O and *username* is the user name for the host object and is used in one-way authentication for iSCSI host logins. If this parameter is not specified, the IQN for the host is used as the user name. To list the CHAP secret and the user name for each host, use the **lsiscsiauth** command. To clear any previously set CHAP secret for a host, use the **chhost -nochapsecret** command.

What to do next

After you create the host object on the system, you can map volumes to a host.

If you are unable to discover the disk on the host system or if there are fewer paths available for each disk than expected, test the connectivity between your host system and the system. Depending on the connection type to the host, these steps might be different. For iSCSI-attached hosts, test your connectivity between the host and system ports by pinging the system from the host. Ensure that the firewall and router settings are configured correctly and validate that the values for the subnet mask and gateway are specified correctly for the system host configuration.

For Fibre Channel attached hosts, ensure that the active switch configuration includes the host zone and check the host-port link status. To verify end-to-end connectivity, you can use the **lsfabric** CLI command or the View Fabric panel under the Service and Maintenance container in the management GUI.

Creating host mappings by using the CLI

You can use the command-line interface (CLI) to create volume-to-host mappings (host mappings).

About this task

Note: To understand the CLI for creating shared mappings in a host cluster, see the information about host clusters and the **mkhostcluster** command.

To create host mappings, follow these steps:

Procedure

1. Issue the **mkvdiskhostmap** CLI command to create host mappings.

   This example is a CLI command that you can issue to create host mappings:

   ```
   mkvdiskhostmap -host demohost1 mainvdisk1
   ```

   Where *demohost1* is the name of the host and *mainvdisk1* is the name of the volume.
2. After you map volumes to hosts, discover the disks on the host system. This step requires that you access the host system and use the host system utilities to discover the new disks that are made available by the system. You also have the option of creating a file system for those new disks. For more information about completing this task, see your host system documentation.

Creating FlashCopy mappings by using the CLI

You can use the command line interface (CLI) to create FlashCopy mappings.

Before you begin

A FlashCopy mapping specifies the source and target volume. Source volumes and target volumes must meet these requirements:

- They must be the same size.
- They must be managed by the same clustered system (system).

About this task

A volume can be the source in up to 256 mappings. A mapping is started at the point in time when the copy is required.

This task creates FlashCopy mappings:

Procedure

1. The source and target volume must be the exact same size. Issue the **lsvdisk -bytes** CLI command to find the size (capacity) of the volume in bytes.
2. Issue the **mkfcmap** CLI command to create a FlashCopy mapping.

   This CLI command example creates a FlashCopy mapping and sets the copy rate:

   ```
   mkfcmap -source mainvdisk1 -target bkpvdisk1
    -name main1copy -copyrate 75
   ```

   Where *mainvdisk1* is the name of the source volume, *bkpvdisk1* is the name of the volume that you want to make the target volume, *main1copy* is the name that you want to call the FlashCopy mapping, and *75* is the copy rate (which translates to MB per second).

   This is an example of the CLI command you can issue to create FlashCopy mappings without the copy rate parameter:

   ```
   mkfcmap -source mainvdisk2 -target bkpvdisk2
    -name main2copy
   ```

   Where *mainvdisk2* is the name of the source volume, *bkpvdisk2* is the name of the volume that you want to make the target volume, *main2copy* is the name that you want to call the FlashCopy mapping.

Note: The default copy rate of 50 (which translates to 2 MB per second) is used when you do not specify a copy rate.

If the specified source and target volumes are also the target and source volumes of an existing mapping, the mapping that is being created and the existing mapping become partners. If one mapping is created as incremental, its partner is automatically incremental. A mapping can have only one partner.

3. Issue the **lsfcmap** CLI command to check the attributes of the FlashCopy mappings that were created:

   This is an example of a CLI command that you can issue to view the attributes of the FlashCopy mappings:

   ```
   lsfcmap -delim :
   ```

   Where **-delim** specifies the delimiter and is an example of the output that is displayed:

   ```
   id:name:source_vdisk_id:source_vdisk_name:target_vdisk_id:target_vdisk_name:
   group_id:group_name:status:progress:copy_rate:clean_progress:incremental
   0:main1copy:77:vdisk77:78:vdisk78:::idle_or_copied:0:75:100:off
   1:main2copy:79:vdisk79:80:vdisk80:::idle_or_copied:0:50:100:off
   ```

Preparing and starting a FlashCopy mapping by using the CLI

Before you start the FlashCopy process by using the command line interface (CLI), you must prepare a FlashCopy mapping.

About this task

Starting a FlashCopy mapping creates a point-in-time copy of the data on the source volume and writes it to the target volume for the mapping.

These steps help you prepare and start a FlashCopy mapping:

Procedure

1. Issue the **prestartfcmap** CLI command to prepare the FlashCopy mapping.

   To run the following command, the FlashCopy mapping cannot belong to a consistency group.

   ```
   prestartfcmap -restore main1copy
   ```

   Where *main1copy* is the name of the FlashCopy mapping.

   This command specifies the optional **restore** parameter, which forces the mapping to be prepared even if the target volume is being used as a source in another active FlashCopy mapping.

   The mapping enters the preparing state and moves to the prepared state when it is ready.

2. Issue the **lsfcmap** CLI command to check the state of the mapping.

   The following code is an example of the output that is displayed:

   ```
   lsfcmap -delim :
   id:name:source_vdisk_id:source_vdisk_name:target_vdisk_id:
   target_vdisk_name:group_id:group_name:status:progress:copy_rate
   0:main1copy:0:mainvdisk1:1:bkpvdisk1:::prepared:0:50
   ```

3. Issue the **startfcmap** CLI command to start the FlashCopy mapping.

   The following code is an example of the CLI command you can issue to start the FlashCopy mapping:

   ```
   startfcmap -restore main1copy
   ```

   Where *main1copy* is the name of the FlashCopy mapping.

   This command specifies the optional **restore** parameter, which forces the mapping to be started even if the target volume is being used as a source in another active FlashCopy mapping.

4. Issue the **lsfcmapprogress** CLI command with the FlashCopy mapping name or ID to check the progress of the mapping.

The following code is an example of the output that is displayed; the FlashCopy mapping ID 0 is 47% completed.

```
lsfcmapprogress -delim :
id:progress
0:47
```

Results

You created a point-in-time copy of the data on a source volume and wrote that data to a target volume. The data on the target volume is only recognized by the hosts that are mapped to it.

Stopping FlashCopy mappings by using the CLI

You can use the command-line interface (CLI) to stop a FlashCopy mapping.

About this task

Follow these steps to stop a single stand-alone FlashCopy mapping.

Procedure

1. To stop a FlashCopy mapping, issue the following **stopfcmap** command:

   `stopfcmap` *`fc_map_id or fc_map_name`*

   where *fc_map_id or fc_map_name* is the ID or name of the mapping to stop.
2. To stop immediately all processing that is associated with the mapping and break the dependency on the source volume of any mappings that are also dependent on the target disk, issue the following command:

   `stopfcmap -force -split` *`fc_map_id or fc_map_name`*

   When you use the **force** parameter, all FlashCopy mappings that depend on this mapping (as listed by the **lsfcmapdependentmaps** command) are also stopped.

   Important: Using the **force** parameter might result in a loss of access. Use it only under the direction of the IBM Support Center.

   The **split** parameter can be specified only when stopping a map that has a progress of 100 as shown by the **lsfcmap** command. The **split** parameter removes the dependency of any other mappings on the source volume. It might be used before starting another FlashCopy mapping whose target disk is the source disk of the mapping that is being stopped. After the mapping is stopped with the **split** option, you can start the other mapping without the **restore** option.

Deleting a FlashCopy mapping using the CLI

You can use the command-line interface (CLI) to delete a FlashCopy mapping.

Before you begin

The **rmfcmap** CLI command deletes an existing mapping if the mapping is in the idle_or_copied or stopped state. If it is in the stopped state, the **force** parameter is required to specify that the target volume is brought online. If the mapping is in any other state, you must stop the mapping before you can delete it.

If deleting the mapping splits the tree that contains the mapping, none of the mappings in either resulting tree can depend on any mapping in the other tree. To display a list of dependent FlashCopy mappings, use the **lsfcmapdependentmaps** command.

About this task

Procedure

1. To delete an existing mapping, issue the **rmfcmap** CLI command:

   `rmfcmap` *`fc_map_id or fc_map_name`*

   where *fc_map_id or fc_map_name* is the ID or name of the mapping to delete.
2. To delete an existing mapping and bring the target volume online, issue the following command:

   `rmfcmap -force` *`fc_map_id or fc_map_name`*

   where *fc_map_id or fc_map_name* is the ID or name of the mapping to delete.

Results

The command does not return any output.

Creating a FlashCopy consistency group and adding mappings using the CLI

You can use the command-line interface (CLI) to create and add mappings to a FlashCopy consistency group.

About this task

If you have created several FlashCopy mappings for a group of volumes that contain elements of data for the same application, it can be convenient to assign these mappings to a single FlashCopy consistency group. You can then issue a single prepare or start command for the whole group. For example, you can copy all of the files for a database at the same time.

Procedure

To add FlashCopy mappings to a new FlashCopy consistency group, complete the following steps.

1. Issue the **mkfcconsistgrp** CLI command to create a FlashCopy consistency group.

   The following CLI command is an example of the command you can issue to create a FlashCopy consistency group:

   ```
   mkfcconsistgrp -name FCcgrp0 -autodelete
   ```

   Where `FCcgrp0` is the name of the FlashCopy consistency group. The **-autodelete** parameter specifies to delete the consistency group when the last FlashCopy mapping is deleted or removed from the consistency group.
2. Issue the **lsfcconsistgrp** CLI command to display the attributes of the group that you have created.

   The following CLI command is an example of the command you can issue to display the attributes of a FlashCopy consistency group:

   ```
   lsfcconsistgrp -delim : FCcgrp0
   ```

   The following output is an example of the output that is displayed:

   ```
   id:1
   name:FCcgrp0
   status:idle_or_copied
   autodelete:on
   FC_mapping_id:0
   FC_mapping_name:fcmap0
   FC_mapping_id:1
   FC_mapping_name:fcmap1
   ```

   Note: For any group that has just been created, the status reported is `empty`

3. Issue the **chfcmap** CLI command to add FlashCopy mappings to the FlashCopy consistency group:

   The following CLI commands are examples of the commands you can issue to add Flash Copy mappings to the FlashCopy consistency group:

   ```
   chfcmap -consistgrp FCcgrp0 main1copy
   chfcmap -consistgrp FCcgrp0 main2copy
   ```

   Where **FCcgrp0** is the name of the FlashCopy consistency group and *main1copy, main2copy* are the names of the FlashCopy mappings.

4. Issue the **lsfcmap** CLI command to display the new attributes of the FlashCopy mappings.

   The following output is an example of the output that is displayed:

   ```
   lsfcmap -delim :
   id:name:source_vdisk_id:source_vdisk_name:target_vdisk_id:
   target_vdisk_name:group_id:group_name:status:progress:copy_rate
   0:main1copy:28:maindisk1:29:bkpdisk1:1:FCcgrp0:idle_copied::75
   1:main2copy:30:maindisk2:31:bkpdisk2:1:FCcgrp0:idle_copied::50
   ```

5. Issue the **lsfcconsistgrp** CLI command to display the detailed attributes of the group.

   The following CLI command is an example of the command that you can issue to display detailed attributes:

   ```
   lsfcconsistgrp -delim : FCcgrp0
   ```

   Where **FCcgrp0** is the name of the FlashCopy consistency group, and **-delim** specifies the delimiter.

   The following output is an example of the output that is displayed:

   ```
   id:1
   name:FCcgrp0
   status:idle_or_copied
   autodelete:off
   FC_mapping_id:0
   FC_mapping_name:main1copy
   FC_mapping_id:1
   FC_mapping_name:main2copy
   ```

Preparing and starting a FlashCopy consistency group using the CLI

You can use the command-line interface (CLI) to prepare and start a FlashCopy consistency group to start the FlashCopy process.

About this task

The successful completion of the FlashCopy process creates a point-in-time copy of the data on the source virtual disk or VDisk (volume) and writes it to the target volume for each mapping in the group. When several mappings are assigned to a FlashCopy consistency group, only a single prepare command is issued to prepare every FlashCopy mapping in the group; only a single start command is issued to start every FlashCopy mapping in the group.

Procedure

To prepare and start a FlashCopy consistency group, complete the following steps.

1. Issue the **prestartfcconsistgrp** CLI command to prepare the FlashCopy consistency group. This command must be issued before the copy process can begin.

   Remember: A single prepare command prepares all of the mappings simultaneously for the entire group.

   An example of the CLI command issued to prepare the FlashCopy consistency group:

   ```
   prestartfcconsistgrp -restore maintobkpfcopy
   ```

   Where *maintobkpfcopy* is the name of the FlashCopy consistency group

The optional **restore** parameter forces the consistency group to be prepared, even if the target volume is being used as a source volume in another active mapping. An active mapping is in the copying, suspended, or stopping state. The group enters the preparing state, and then moves to the prepared state when it is ready.

2. Issue the **lsfcconsistgrp** command to check the status of the FlashCopy consistency group.

   An example of the CLI command issued to check the status of the FlashCopy consistency group.

   ```
   lsfcconsistgrp -delim :
   ```

   An example of the output displayed:

   ```
   id:name:status
   1:maintobkpfcopy:prepared
   ```

3. Issue the **startfcconsistgrp** CLI command to start the FlashCopy consistency group to make the copy.

   Remember: A single start command starts all the mappings simultaneously for the entire group.

   An example of the CLI command issued to start the FlashCopy consistency group mappings:

   ```
   startfcconsistgrp -prep -restore maintobkpfcopy
   ```

   Where *maintobkpfcopy* is the name of the FlashCopy consistency group

   Include the **prep** parameter, and the system automatically issues the **prestartfcconsistgrp** command for the specified group.

   Note: Combining the **restore** parameter with the **prep** parameter, force-starts the consistency group. This occurs even if the target volume is being used as a source volume in another active mapping. An active mapping is in the copying, suspended, or stopping state.
   The FlashCopy consistency group enters the copying state and returns to the idle_copied state when complete.

4. Issue the **lsfcconsistgrp** command to check the status of the FlashCopy consistency group.

   An example of the CLI command issued to check the status of the FlashCopy consistency group:

   ```
   lsfcconsistgrp -delim : maintobkpfcopy
   ```

   Where *maintobkpfcopy* is the name of the FlashCopy consistency group

   An example of the output that is displayed during the copying process:

   ```
   id:name:status
   1:maintobkpfcopy:copying
   ```

   An example of the output that is displayed when the process copying is complete:

   ```
   id:1
   name:maintobkpfcopy
   status:idle_copied
   autodelete:off
   FC_mapping_id:0
   FC_mapping_name:main1copy
   FC_mapping_id:1
   FC_mapping_name:main2copy
   ```

Stopping a FlashCopy consistency group using the CLI

You can use the command-line interface (CLI) to stop a FlashCopy consistency group.

Before you begin

The **stopfcconsistgrp** CLI command stops all processing that is associated with a FlashCopy consistency group that is in one of the following processing states: prepared, copying, stopping, or suspended.

About this task

Procedure

1. To stop a FlashCopy consistency group, issue the **stopfcconsistgrp** CLI command:

   ```
   stopfcconsistgrp fc_map_id or fc_map_name
   ```

   where *fc_map_id or fc_map_name* is the ID or name of the mapping to delete.
2. To stop a consistency group and break the dependency on the source volumes of any mappings that are also dependent on the target volume, issue the following command:

   ```
   stopfcconsistgrp -split fc_map_id or fc_map_name
   ```

   You can specify the **split** parameter when all the maps in the group have a progress of 100. It removes the dependency of any other maps on the source volumes. You can use this option before you start another FlashCopy consistency group whose target disks are the source disks of the mappings that are being stopped. After the consistency group is stopped with the split option, you can start the other consistency group without the restore option.

Results

The command does not return any output.

Deleting a FlashCopy consistency group using the CLI

You can use the command-line interface (CLI) to delete a FlashCopy consistency group.

Before you begin

The **rmfcconsistgrp** CLI command deletes an existing FlashCopy consistency group. The **-force** parameter is required only when the consistency group that you want to delete contains mappings.

About this task

Follow these steps to delete an existing consistency group:

Procedure

1. To delete an existing consistency group that does not contain mappings, issue the **rmfcconsistgrp** CLI command:

   ```
   rmfcconsistgrp fc_map_id or fc_map_name
   ```

   where *fc_map_id or fc_map_name* is the ID or name of the consistency group to delete.
2. To delete an existing consistency group that contains mappings that are members of the consistency group, issue the following command:

   ```
   rmfcconsistgrp -force fc_map_id or fc_map_name
   ```

   where *fc_map_id or fc_map_name* is the ID or name of the mapping to delete.

   Important: Using the **-force** parameter might result in a loss of access. Use it only under the direction of your support center.

   All the mappings that are associated with the consistency group are removed from the group and changed to stand-alone mappings. To delete a single mapping in the consistency group, you must use the **rmfcmap** command.

Results

The command does not return any output.

Creating Metro Mirror, Global Mirror, or active-active relationships by using the CLI

You can use the command-line interface (CLI) to create Metro Mirror, Global Mirror, or active-active relationships.

About this task

Complete these steps to create Metro Mirror, Global Mirror, or active-active relationships:

Procedure

1. To create a Metro Mirror relationship, run the **mkrcrelationship** command. For example, enter:

   ```
   mkrcrelationship -master master_volume_id
   -aux aux_volume_id -cluster system_id
   ```

   Where *master_volume_id* is the ID of the master volume, *aux_volume_id* is the ID of the auxiliary volume, and *system_id* is the ID of the remote clustered system.
2. To create a new Global Mirror relationship, run the **mkrcrelationship** command with the **-global** parameter. For example, enter:

   ```
   mkrcrelationship -master master_volume_id
   -aux aux_volume_id -cluster system_id -global
   ```

   Where *master_volume_id* is the ID of the master volume, *aux_volume_id* is the ID of the auxiliary volume, and *system_id* is the ID of the remote system.
3. To create a new relationship with cycling enabled:

   ```
   mkrcrelationship -master books_volume -aux books_volume -cluster DR_cluster -global -cyclingmode multi
   ```

   Note: Add change volumes to a relationship by issuing **`chrcrelationship -auxchange`** or **`chrcrelationship -masterchange`**.
4. To create a new active-active relationship, run the **mkrcrelationship** command with the **-activeactive** parameter. For example, enter the following command:

   ```
   mkrcrelationship -master master_volume_id -aux aux_volume_id -cluster system_id -activeactive
   ```

   Where *master_volume_id* is the ID of the master volume, *aux_volume_id* is the ID of the auxiliary volume, and *system_id* is the ID of the remote system.

Modifying Metro Mirror, Global Mirror, or active-active relationships by using the CLI

You can use the command-line interface (CLI) to modify certain attributes of Metro Mirror, Global Mirror, or active-active relationships. You can change only one attribute at a time for each command submission.

About this task

To modify Metro Mirror, Global Mirror, or active-active relationships, run the **chrcrelationship** command.

Procedure

Run the **chrcrelationship** command to change the name of a Metro Mirror, Global Mirror, or active-active relationship. For example, to change the relationship name, enter:

```
chrcrelationship -name new_rc_rel_name previous_rc_rel_name
```

Where *new_rc_rel_name* is the new name of the relationship and *previous_rc_rel_name* is the previous name of the relationship.
Or, run the **chrcrelationship** command to remove a relationship from whichever consistency group it is a member of. For example, enter the following command:

```
chrcrelationship -force -noconsistgrp rc_rel_name/id
```

Where *rc_rel_name/id* is the name or ID of the relationship.

Important: Using the **-force** parameter might result in a loss of access. Use it only under the direction of your support center.

Starting and stopping Metro Mirror, Global Mirror, or active-active relationships by using the CLI

You can use the command-line interface (CLI) to start and stop stand-alone Metro Mirror, Global Mirror, or active-active relationships. Relationships that are members of consistency groups must be started and stopped by using the consistency group CLI commands.

About this task

Complete these steps to start or stop Metro Mirror, Global Mirroror active-active relationships:

Procedure

1. To start a Metro Mirror,Global Mirror, or an active-active relationship, run the **startrcrelationship** command. For example, enter the following command:

   ```
   startrcrelationship rc_rel_id
   ```

   Where *rc_rel_id* is the ID of the relationship that you want to start in a stand-alone relationship.

   Note: Active-active relationships can be started only if they have a state of idling.
2. To stop a Metro Mirror or Global Mirror relationship, run the **stoprcrelationship** command. This command applies to a stand-alone relationship.

   For example, enter the following command:

   ```
   stoprcrelationship rc_rel_id
   ```

   Where *rc_rel_id* is the ID of the stand-alone relationship that you want to stop mirroring I/O.
3. To stop an active-active relationship, the following conditions must be met:
   - The **-access** parameter is specified.
   - The state of the relationship is **consistent_copying**.
   - The status of the relationship is **primary_offline**.

   For example, enter the following command:

   ```
   stoprcrelationship rc_rel_id -access
   ```

   Where *rc_rel_id* is the ID of the active-active relationship that you want to stop. The **-access** parameter gives hosts read or write access to a volume in an active-active relationship that contains an older but a consistent image that can be used in a disaster recovery scenario.

Displaying the progress of Metro Mirror, Global Mirror, or active-active relationships by using the CLI

You can use the command-line interface (CLI) to display the background copy of Metro Mirror, Global Mirror, or active-active relationships as a percentage. When the initial background copy process for a relationship is completed, `null` is displayed for the progress of that relationship.

About this task

To display the progress of the background copy of Metro Mirror, Global Mirror, or active-active relationships, run the **lsrcrelationshipprogress** command.

Procedure

1. To display data progress without headings for columns of data or for each item of data in a Metro Mirror, Global Mirror, or active-active relationship, run the **lsrcrelationshipprogress -nohdr** command. For example, to display data of the relationship with headings suppressed, enter the following command, where *rc_rel_name* is the name of the specified object type.

   ```
   lsrcrelationshipprogress -nohdr rc_rel_name
   ```

2. To display the progress of a background copy of a Metro Mirror, Global Mirror, or active-active relationship as a percentage, run the **lsrcrelationshipprogress -delim** command. The colon character (:) separates all items of data in a concise view, and the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter. For example, enter the following command:

   ```
   lsrcrelationshipprogress -delim : 0
   ```

   The resulting output is displayed, such as in this example:

   ```
   id:progress
   0:58
   ```

Switching Metro Mirror or Global Mirror relationships using the CLI

You can use the command-line interface (CLI) to reverse the roles of primary and secondary volumes in a stand-alone Metro Mirror or Global Mirror relationship when that relationship is in a consistent state. You cannot switch roles between primary and secondary volumes for active-active relationships.

About this task

Relationships that are members of consistency groups must be switched by using the consistency group CLI commands. To switch the roles of primary and secondary volumes in Metro Mirror or Global Mirror relationships, follow these steps:

Procedure

1. To make the master disk in a Metro Mirror or Global Mirror relationship to be the primary, run the **switchrcrelationship -primary master** command. For example, enter:

   ```
   switchrcrelationship -primary master rc_rel_id
   ```

   Where *rc_rel_id* is the ID of the relationship to switch.

2. To make the auxiliary disk in a Metro Mirror or Global Mirror relationship to be the primary, run the **switchrcrelationship -primary aux** command. For example, enter:

   ```
   switchrcrelationship -primary aux rc_rel_id
   ```

   Where *rc_rel_id* is the ID of the relationship to switch.

   Remember:

   - You cannot switch a global relationship if cycling is (automatically) set.
   - To switch the direction of a **multi** cycling mode-based relationship, the relationship must stop with access enabled. Then, start by using **`-force`** in the opposite direction. (Using the **`force`** parameter might result in a loss of access. Use it only under the direction of your support center.)

Deleting Metro Mirror, Global Mirror, or active-active relationships by using the CLI

You can use the command line interface (CLI) to delete Metro Mirror, Global Mirror, or active-active relationships.

Procedure

To delete Metro Mirror, Global Mirror, or active-active relationships, run the **rmrcrelationship** command. For example, enter the following command:

```
rmrcrelationship rc_rel_name/id
```

Where *rc_rel_name/id* is the name or ID of the relationship.

Creating Metro Mirror, Global Mirror, or active-active consistency groups by using the CLI

You can use the command-line interface (CLI) to create Metro Mirror, Global Mirror, or active-active consistency groups.

About this task

To create Metro Mirror, Global Mirror, or active-active consistency groups, complete these steps:

Procedure

1. To create a Metro Mirror, Global Mirror, or active-active consistency group, run the **mkrcconsistgrp** command. For example, enter the following command:

   ```
   mkrcconsistgrp -name new_name -cluster cluster_id
   ```

   where *new_name* is the name of the new consistency group and *cluster_id* is the ID of the remote cluster for the new consistency group. If **-cluster** is not specified, a consistency group is created only on the local cluster. The new consistency group does not contain any relationships and is in the empty state.
2. To add Metro Mirror, Global Mirror, or active-active relationships to the group, run the **chrcrelationship** command. For example, enter the following command:

   ```
   chrcrelationship -consistgrp consist_group_name rc_rel_id
   ```

   where *consist_group_name* is the name of the new consistency group to assign the relationship to and *rc_rel_id* is the ID of the relationship.

Modifying Metro Mirror, Global Mirror, or active-active consistency groups by using the CLI

You can use the command-line interface (CLI) to assign a new name or modify the name of an existing Metro Mirror, Global Mirror, or active-active consistency group.

About this task

To assign or modify the name of a Metro Mirror, Global Mirror, or active-active consistency group, run the **chrcconsistgrp** command.

Procedure

1. Run the **chrcconsistgrp** command to assign a new name to the consistency group. For example, enter the following command:

   ```
   chrcconsistgrp -name new_name_arg
   ```

   where *new_name_arg* is the assigned new name of the consistency group.
2. Run the **chrcconsistgrp** command to change the name of the consistency group. For example, enter the following command:

   ```
   chrcconsistgrp -name new_consist_group_name previous_consist_group_name
   ```

   where *new_consist_group_name* is the assigned new name of the consistency group and *previous_consist_group_name* is the previous name of the consistency group.

Starting and stopping Metro Mirror, Global Mirror, or active-active consistency-group copy processes by using the CLI

You can use the command-line interface (CLI) to start and stop Metro Mirror, Global Mirror, or active-active consistency-group copy processes.

About this task

To start and stop Metro Mirror, Global Mirror, or active-active consistency-group copy processes, complete these steps:

Procedure

1. To start a Metro Mirror, Global Mirror, or active-active consistency-group copy process, set the direction of copy if it is undefined and optionally mark the secondary volumes of the consistency group as clean. Run the **startrcconsistgrp** command. For example, enter the following command:

   ```
   startrcconsistgrp rc_consist_group_id
   ```

   where *rc_consist_group_id* is the ID of the consistency group to start processing.

   Note: If you are starting an active-active consistency group, all relationships in the group must be in **idling** state for the consistency group to start.
2. To stop the copy process for a Metro Mirror or Global Mirror consistency group, run the **stoprcconsistgrp** command.

   For example, enter the following command:

   ```
   stoprcconsistgrp rc_consist_group_id
   ```

   Where *rc_consist_group_id* is the ID of the consistency group that you want to stop processing.

   If the group is in a consistent state, you can also use this command to enable write access to the secondary volumes in the group.
3. To stop the copy process for an active-active consistency group, the following conditions must be met:
   - The **-access** parameter is specified.
   - The state of the relationships in the consistency group is **consistent_copying**.
   - The status of the relationships in the consistency group is **primary_offline**.

   For example, enter the following command:

   ```
   stoprcconsistgrp rc_consist_group_id -access
   ```

   Where *rc_rel_id* is the ID of the active-active consistency group that you want to stop. The **-access** parameter gives hosts read or write access to a volume in an active-active relationship that contains an older but a consistent image that can be used in a disaster recovery scenario.

Deleting Metro Mirror, Global Mirror, or active-active consistency groups by using the CLI

You can use the command-line interface (CLI) to delete Metro Mirror, Global Mirror, or active-active consistency groups.

About this task

To delete existing Metro Mirror, Global Mirror, or active-active consistency groups, complete these steps:

Procedure

1. To delete a Metro Mirror, Global Mirror, or active-active consistency group, run the **rmrcconsistgrp** command. For example, enter the following command:

   ```
   rmrcconsistgrp rc_consist_group_id
   ```

   Where *rc_consist_group_id* is the ID of the consistency group to delete.

2. If a Metro Mirror, Global Mirror, or active-active consistency group is not empty, you must use the **-force** parameter to delete the consistency group. For example, enter the following command:

   ```
   rmrcconsistgrp -force rc_consist_group_id
   ```

   Where *rc_consist_group_id* is the ID of the consistency group to delete. This command causes all relationships that are members of the deleted group to become stand-alone relationships.

   Important: Using the **force** parameter might result in a loss of access. Use it only under the direction of your support center.

Creating Metro Mirror and Global Mirror partnerships by using the CLI

You can use the command-line interface (CLI) to create Metro Mirror and Global Mirror partnerships between two clusters.

Procedure

Note: When remote copy partnerships are created between systems that support different maximum numbers of volumes, then the maximum number of volumes that can be created on any system is determined to be the same as on the system that supports the smallest maximum number of volumes. If one system has more disks than is supported by the other system, the attempt to create a partnership fails.
To create Metro Mirror and Global Mirror partnerships, complete the following steps.

1. To create Metro Mirror and Global Mirror partnerships for Fibre Channel connections, run the **mkfcpartnership** command. To create Metro Mirror and Global Mirror partnerships for IP connections, run the **mkippartnership** command. For example, for Fibre Channel connections enter the following command:

   ```
   mkfcpartnership -linkbandwidthmbits bandwidth_in_mbps
   -backgroundcopyrate percentage_of_available_bandwidth remote_cluster_id
   ```

   where *bandwidth_in_mbps* specifies the bandwidth (in megabytes per second) that is used by the background copy process between the clusters, *percentage_of_available_bandwidth* specifies the maximum percentage of aggregate link bandwidth that can be used for background copy operations, and *remote_cluster_id* is the ID of the remote system. For IP connections, enter the following command:

   ```
   mkippartnership -type ip_address_type
                   -clusterip remote_cluster_ip_address
                   -chapsecret chap_secret
                   -linkbandwidthmbits bandwidth_in_mbps
                   -backgroundcopyrate percentage_of_available_bandwidth
   ```

   where *ip_address_type* specifies the IP address type (IPv4 or IPv6) that is used by the background copy process between the clusters, *remote_cluster_ip_address* specifies the IP address of the remote system, *chap_secret* specifies the CHAP secret of the remote system (optional), *bandwidth_in_mbps* specifies the bandwidth (in megabytes per second) that is used by the background copy process between the clusters, and *percentage_of_available_bandwidth* specifies the maximum percentage of aggregate link bandwidth that can be used for background copy operations (optional).
2. Run the **mkfcpartnership** command for Fibre Channel connections or **mkippartnership** command for IP connections from the remote system. For example, for Fibre Channel connections enter the following command:

   ```
   mkfcpartnership -linkbandwidthmbits bandwidth_in_mbps
                   -backgroundcopyrate percentage_of_available_bandwidth
                   partner_cluster_id
   ```

   where *bandwidth_in_mbps* specifies the bandwidth (in megabytes per second) that is used by the background copy process between the clusters, *percentage_of_available_bandwidth* specifies the maximum percentage of aggregate link bandwidth that can be used for background copy operations, and *partner_cluster_id* is the ID of the partner system (the local system in the previous step).

   For Internet Protocol (IP) connections, enter the following command:

```
mkippartnership -type ip_address_type
                -clusterip partner_cluster_ip_address
                -chapsecret chap_secret
                -linkbandwidthmbits bandwidth_in_mbps
                -backgroundcopyrate percentage_of_available_bandwidth
```

where *ip_address_type* specifies the IP address type (IPv4 or IPv6) that is used by the background copy process between the clusters, *partner_cluster_ip_address* specifies the IP address of the partner system, *chap_secret* specifies the CHAP secret of the partner system (optional), *bandwidth_in_mbps* specifies the bandwidth (in megabytes per second) that is used by the background copy process between the clusters, and *percentage_of_available_bandwidth* specifies the maximum percentage of aggregate link bandwidth that can be used for background copy operations (optional). The partner system is the local system from the previous step.

Modifying Metro Mirror and Global Mirror partnerships using the CLI

You can use the command-line interface (CLI) to modify Metro Mirror and Global Mirror partnerships.

About this task

The partnership bandwidth, which is also known as *background copy*, controls the rate at which data is sent from the local system to the remote clustered system (system). The partnership bandwidth can be changed to help manage the use of intersystem links. It is measured in megabytes per second (MBps).

Complete the following steps to modify Metro Mirror and Global Mirror partnerships:

Procedure

1. To modify Metro Mirror and Global Mirror partnerships, run the **chpartnership** command. For example, enter:

   ```
   chpartnership -type ip_address_type
                 -clusterip remote_cluster_ip_address
                 -chapsecret chap_secret
                 -nochapsecret -linkbandwidthmbits bandwidth_in_mbps
                 -backgroundcopyrate percentage_of_available_bandwidth remote_cluster_id
   ```

   where *ip_address_type* specifies the IP address type ("ipv4" or "ipv6") that is used by the background copy process between the clusters (only used for IP connections), *remote_cluster_ip_address* specifies the IP address of the remote cluster (only used for IP connections), *chap_secret* specifies the CHAP secret of the remote cluster (only used for IP connections), *bandwidth_in_mbps* specifies the bandwidth (in megabytes per second) that is used by the background copy process between the clusters (this is optional), *percentage_of_available_bandwidth* specifies the maximum percentage of aggregate link bandwidth that can be used for background copy operations (this is optional), and *remote_cluster_id* is the ID or name of the remote system.
2. Run the **chpartnership** command from the remote system. For example, enter:

   ```
   chpartnership -type ip_address_type
                 -clusterip local_cluster_ip_address
                 -chapsecret chap_secret -nochapsecret
                 -linkbandwidthmbits bandwidth_in_mbps
                 -backgroundcopyrate percentage_of_available_bandwidth local_cluster_id
   ```

   where *ip_address_type* specifies the IP address type ("ipv4" or "ipv6") that is used by the background copy process between the clusters (only used for IP connections), *local_cluster_ip_address* specifies the IP address of the local cluster (only used for IP connections), *chap_secret* specifies the CHAP secret of the local cluster (only used for IP connections), *bandwidth_in_mbps* specifies the bandwidth (in megabytes per second) that is used by the background copy process between the clusters (this is optional), *percentage_of_available_bandwidth* specifies the maximum percentage of aggregate link bandwidth that can be used for background copy operations (this is optional), and *local_cluster_id* is the ID or name of the local system.

Starting and stopping Metro Mirror and Global Mirror partnerships using the CLI

You can use the command-line interface (CLI) to start and stop Metro Mirror and Global Mirror partnerships.

About this task

Complete the following steps to start and stop Metro Mirror and Global Mirror partnerships:

Procedure

1. To start a Metro Mirror or Global Mirror partnership, run the **chpartnership** command from either cluster. For example, enter:

   ```
   chpartnership -start cluster_id
   ```

   Where *cluster_id* is the ID of the local or remote cluster. The **mkfcpartnership** or **mkippartnership** command starts the partnership by default.
2. To stop a Metro Mirror or Global Mirror partnership, run the **chpartnership** command from either cluster.

   For example, enter:

   ```
   chpartnership -stop cluster_id
   ```

   Where *cluster_id* is the ID of the local or remote cluster.

Deleting Metro Mirror and Global Mirror partnerships using the CLI

You can use the command-line interface (CLI) to delete Metro Mirror and Global Mirror partnerships.

About this task

Complete the following steps to delete Metro Mirror and Global Mirror partnerships:

Procedure

1. If a Metro Mirror or Global Mirror partnership has configured relationships or groups, you must stop the partnership before you can delete it. For example, enter:

   ```
   chpartnership -stop remote_cluster_id
   ```

   Where *remote_cluster_id* is the ID of the remote cluster.
2. To delete a Metro Mirror and Global Mirror partnership, run the **rmpartnership** command from either cluster. For example, enter:

   ```
   rmpartnership remote_cluster_id
   ```

   Where *remote_cluster_id* is the ID of the remote cluster.

Determining the WWNNs of a node using the CLI

You can determine the worldwide node names (WWNNs) of a node using the command-line interface (CLI).

About this task

Perform the following steps to determine the WWNNs of a node:

Procedure

1. Issue the **lsnode** CLI command to list the nodes in the clustered system.
2. Record the name or ID of the node for which you want to determine the WWNNs.
3. Issue the **lsportfc** CLI command and specify the node name or ID that was recorded in step 2.

Here is an example of the CLI command you can issue:

```
lsportfc -filtervalue node_id=2
```

Where `node_id=2` is the name of the node for which you want to determine the WWNNs. The output from the command is:

```
id  fc_io_port_idport_id  type      port_speed node_id node_nameWWNN              nportid status
0   1              1      fc        8 Gb       2       node2    5005076801405F82  010E00  active
1   2              2      fc        8 Gb       2       node2    5005076801305F82  010A00  active
2   3              3      fc        8 Gb       2       node2    5005076801105F82  010E00  active
3   4              4      fc        8 Gb       2       node2    5005076801205F82  10A00   active
4   5              3      ethernet  10 Gb      2       node2    5005076801505F82  540531  active
5   6              4      ethernet  10 Gb      2       node2    5005076801605F82  E80326  active
```

4. Record the six WWNNs (to assist with setting up other systems).

Listing node-dependent volumes using the CLI

You can use the command-line interface (CLI) to list the volumes that are dependent on the status of a node.

Before you begin

If a node goes offline or is removed from a system, all volumes that are dependent on the node go offline. Before taking a node offline or removing a node from a system, run the **lsdependentvdisks** command to identify any node-dependent volumes.

About this task

By default, the **lsdependentvdisks** command also checks all available quorum disks. If the quorum disks are accessible only through the specified node, the command returns an error.

Various scenarios can produce node-dependent volumes. The following examples are common scenarios in which the **lsnodedependentvdisks** command will return node-dependent volumes:

1. The node contains flash drives the only synchronized copy of a mirrored volume.
2. The node is the only node that can access an MDisk on the SAN fabric.
3. The other node in the I/O group is offline (all volumes in the I/O group are returned).
4. Pinned data in the cache is stopping the partner node from joining the I/O group.

To resolve (1), allow volume mirror synchronizations between Flash drive MDisks to complete. To resolve (2-4), bring any offline MDisks online and repair any degraded paths.

Note: The command lists the node-dependent volumes at the time the command is run; subsequent changes to a system require running the command again.

Procedure

1. Issue the **lsdependentvdisks** CLI command.

   The following example shows the CLI format for listing the volumes that are dependent on node01:

   ```
   lsdependentvdisks -enclosure -delim : 0:1
   ```

   The following example shows the output that is displayed:

   ```
   vdisk_id:vdisk_name
   4:vdisk4
   5:vdisk5
   ```

2. If the **lsdependentvdisks** command returns an error, you must move your quorum disks to MDisks that are accessible through all nodes. Rerun the command until no errors are returned.

3. Reissue the **lsdependentvdisks** command. When the command returns no volumes, the system is free from any node-dependent volumes.

   The following example shows the command syntax for listing the volumes that are dependent on `node01`:

   ```
   lsdependentvdisks -delim : -node node01 :
   ```

   The following example shows the command output if there are no node-dependent volumes in the system:

   ```
   vdisk_id       vdisk_name
   ```

Determining the volume name from the device identifier on the host

You can use the command-line interface (CLI) to determine the volume name from the device identifier on the host.

About this task

Each volume that is exported by the system is assigned a unique device identifier. The device identifier uniquely identifies the volume and can be used to determine which volume corresponds to the volume that the host detects.

Complete the following steps to determine the volume name from the device identifier:

Procedure

1. Find the device identifier. For example, if you are using the subsystem device driver (SDD), the disk identifier is referred to as the virtual path (vpath) number. You can issue the following SDD command to find the vpath serial number:

   ```
   datapath query device
   ```

   For other multipathing drivers, refer to the documentation that is provided with your multipathing driver to determine the device identifier.
2. Find the host object that is defined to the system and corresponds with the host that you are working with.
   a. Find the worldwide port numbers (WWPNs) by looking at the device definitions that are stored by your operating system. For example, on AIX® the WWPNs are in the ODM and if you use Windows you must go into the HBA BIOS.
   b. Verify which host object is defined to the system for which these ports belong. The ports are stored as part of the detailed view so that you must list each host by issuing the following CLI command:

      ```
      lshost id | name
      ```

      Where *id | name* is the name or ID of the host.
   c. Check for matching WWPNs.
3. Enter the following command to list the host mappings:

   ```
   lshostvdiskmap hostname
   ```

   Where *hostname* is the name of the host.
4. Find the volume UID that matches the device identifier and record the volume name or ID.

Determining the host that a volume maps

You can determine the host that a volume maps by using the command-line interface (CLI). To view the host mapping for a volume in the management GUI, select **Volumes** > **Volumes by Hosts**.

About this task

Complete the following steps to determine the host that the volume maps:

Procedure

1. Enter the following CLI command to list the hosts to which this volume maps:
   ```
   lsvdiskhostmap vdisk_name | vdisk_id
   ```
   Where *vdisk_name* | *vdisk_ id* is the name or ID of the volume.
2. Find the host name or ID to determine which host this volume maps.
   - If no data is returned, the volume does not map any hosts.

Determining the relationship between volume and MDisks using the CLI

You can determine the relationship between volumes and managed disks (MDisks) using the command-line interface (CLI).

About this task

Select one or more of the following options to determine the relationship between volumes and MDisks:

Procedure

- To display a list of the IDs that correspond to the MDisks that comprise the volume, issue the following CLI command:
  ```
  lsvdiskmember vdiskname/id
  ```
  where *vdiskname/id* is the name or ID of the volume.
- To display a list of IDs that correspond to the volumes that are using this MDisk, issue the following CLI command:
  ```
  lsmdiskmember mdiskname/id
  ```
  where *mdiskname/id* is the name or ID of the MDisk.
- To display a table of volume IDs and the corresponding number of extents that are being used by each volume, issue the following CLI command:
  ```
  lsmdiskextent mdiskname/id
  ```
  where *mdiskname/id* is the name or ID of the MDisk.
- To display a table of MDisk IDs and the corresponding number of extents that each MDisk provides as storage for the specified volume, issue the following CLI command:
  ```
  lsvdiskextent vdiskname/id
  ```
  where *vdiskname/id* is the name or ID of the volume.

Determining the relationship between MDisks and controller LUNs using the CLI

You can determine the relationship between managed disks (MDisks) and RAID arrays or LUNs using the command-line interface (CLI).

About this task

Each MDisk corresponds with a single RAID array, or with a single partition on a specified RAID array. Each RAID controller defines a LUN number for this disk. The LUN number and controller name or ID are needed to determine the relationship between MDisks and RAID arrays or partitions.

Complete the following steps to determine the relationship between MDisks and RAID arrays:

Procedure

1. Enter the following command to display a detailed view of the MDisk:

   ```
   lsmdisk object_name
   ```

   Where *object_name* is the name of the MDisk for which you want to display a detailed view.
2. Record the controller name or controller ID and the controller LUN number.
3. Enter the following command to display a detailed view of the controller:

   ```
   lscontroller controller_name
   ```

   Where *controller_name* is the name of the controller that you recorded in step 2.
4. Record the vendor ID, product ID, and WWNN. You can use this information to determine what is being presented to the MDisk.
5. From the native user interface for the specified controller, list the LUNs it is presenting and match the LUN number with that noted in step 1. This provides the exact RAID array or partition that corresponds with the MDisk.

Increasing the size of your system by using the CLI

You can increase the size of the system by adding more nodes. The nodes must be added in pairs and assigned to a new I/O group.

About this task

Complete the following steps to increase the size of your system:

Procedure

1. Add a node to your system and repeat this step for the second node.
2. Migrate your volumes to new I/O groups if you want to balance the load between the existing I/O groups and the new I/O groups. Repeat this step for all volumes that you want to assign to the new I/O group.

Adding a node to increase the size of the system

You can add a node to the system by using the CLI or management GUI. A node can be added to the system if the node previously failed and is being replaced with a new node or if a repair action causes the node to be unrecognizable by the system. When you add nodes, ensure that they are added in pairs to create a full I/O group. Adding a node to the system typically increases the capacity of the entire system. Adding spare nodes to a system does not increase the capacity of the system.

You can use either the management GUI or the command-line interface to add a node to the system. Some models might require you to use the front panel to verify that the new node was added correctly.

Before you add a node to a system, you must make sure that the switch zoning is configured such that the node that is being added is in the same zone as all other nodes in the system. If you are replacing a node and the switch is zoned by worldwide port name (WWPN) rather than by switch port, make sure that the switch is configured such that the node that is being added is in the same VSAN or zone.

Note: It is recommended that you use a consistent method (either only the management GUI, or only the CLI) when you add, remove, and re-add nodes. If a node is added by using the CLI and later re-added by using the GUI, it might get a different node name than it originally had.

Rules and restrictions for adding a node to a system

If you are using hot-spare nodes, the following considerations might not all be applicable. For more information, see the topic on adding a hot-spare node and the **swapnode** command.

If you are adding a node that was used previously, either within a different I/O group within this system or within a different system, if you add a node without changing its worldwide node name (WWNN), hosts might detect the node and use it as if it were in its old location. This action might cause the hosts to access the wrong volumes.

- You must ensure that the model type of the new node is supported by the software level that is installed on the system. If the model type is not supported by the software level, update the system to a software level that supports the model type of the new node.
- Each node in an I/O group must be connected to a different uninterruptible power supply.
- If you are adding a node back to the same I/O group after a service action required it to be deleted from the system, and if the physical node did not change, then no special procedures are required to add it back to the system.
- If you are replacing a node in a system either because of a node failure or an update, you must change the WWNN of the new node to match that of the original node before you connect the node to the Fibre Channel network and add the node to the system.
- If you are adding a node to the network again, to avoid data corruption, ensure that you are adding the node to the same I/O group from which it was removed. You must use the information that was recorded when the node was originally added to the system. If you do not have access to this information, contact the support center for assistance with adding the node back into the system so that data is not corrupted.
- For each external storage system, the LUNs that are presented to the ports on the new node must be the same as the LUNs that are presented to the nodes that currently exist in the system. You must ensure that the LUNs are the same before you add the new node to the system.
- If you create an I/O group in the system and add a node, no special procedures are needed because this node was never added to a system.
- If you create an I/O group in the system and add a node that was added to a system before, the host system might still be configured to the node WWPNs and the node might still be zoned in the fabric. Because you cannot change the WWNN for the node, you must ensure that other components in your fabric are configured correctly. Verify that any host that was previously configured to use the node was correctly updated.
- If the node that you are adding was previously replaced, either for a node repair or update, you might use the WWNN of that node for the replacement node. Ensure that the WWNN of this node was updated so that you do not have two nodes with the same WWNN attached to your fabric. Also, ensure that the WWNN of the node that you are adding is not 00000. If it is 00000, contact your support representative.
- The new node must be running a software level that supports encryption.
- If you are adding the new node to a system with either a HyperSwap or stretched system topology, you must assign the node to a specific site.

Rules and restrictions for using multipathing device drivers

- Applications on the host systems direct I/O operations to file systems or logical volumes that are mapped by the operating system to virtual paths (*vpaths*), which are pseudo disk objects that are supported by the multipathing device drivers. Multipathing device drivers maintain an association between a vpath and a volume. This association uses an identifier (UID) which is unique to the volume and is never reused. The UID allows multipathing device drivers to directly associate vpaths with volumes.
- Multipathing device drivers operate within a protocol stack that contains disk and Fibre Channel device drivers that are used to communicate with the system by using the SCSI protocol over Fibre Channel as defined by the ANSI FCS standard. The addressing scheme that is provided by these SCSI

and Fibre Channel device drivers uses a combination of a SCSI logical unit number (LUN) and the worldwide node name (WWNN) for the Fibre Channel node and ports.

- If an error occurs, the error recovery procedures (ERPs) operate at various tiers in the protocol stack. Some of these ERPs cause I/O to be redriven by using the same WWNN and LUN numbers that were previously used.
- Multipathing device drivers do not check the association of the volume with the vpath on every I/O operation that it performs.

You can use either the **addnode** command or the **Add Node** wizard in the management GUI. To access the **Add Node** wizard, select **Monitoring** > **System**. On the image, click the new node to start the wizard. Complete the wizard and verify the new node. If the new node is not displayed in the image, it indicates a potential cabling issue. Check the installation information to ensure that your node was cabled correctly.

To add a node to a system by using the command-line interface, complete these steps:

1. Enter this command to verify that the node is detected on the network:

   ```
   svcinfo lsnodecandidate
   ```

   This example shows the output for this command:

   ```
   # svcinfo lsnodecandidate
   id               panel_name UPS_serial_number UPS_unique_id    hardware serial_number product_mtm machine_signature

   500507680C007B00 KD0N8AM                       500507680C007B00  DH8     KD0N8AM       2145-DH8    0123-4567-89AB-CDEF
   ```

   The **id** parameter displays the WWNN for the node. If the node is not detected, verify cabling to the node.
2. Enter this command to determine the I/O group where the node must be added:

   ```
   lsiogrp
   ```
3. Record the name or ID of the first I/O group that has a node count of zero. You need the name or ID for the next step. Note: You must do this step for the first node that is added. You do not do this step for the second node of the pair because it uses the same I/O group number.
4. Enter this command to add the node to the system:

   ```
   addnode -wwnodename WWNN -iogrp iogrp_name -name new_name_arg -site site_name
   ```

   Where **WWNN** is the WWNN of the node, **iogrp_name** is the name of the I/O group that you want to add the node to and new_name_arg is the name that you want to assign to the node. If you do not specify a new node name, a default name is assigned. Typically, you specify a meaningful node name. The site_name specifies the name of the site location of the new node. This parameter is only required if the topology is a HyperSwap or stretched system.

   Note: Adding the node might take a considerable amount of time.
5. Record this information for future reference:
   - Serial number.
   - Worldwide node name.
   - All of the worldwide port names.
   - The name or ID of the I/O group

Validating and repairing mirrored volume copies by using the CLI

You can use the **repairvdiskcopy** command from the command-line interface (CLI) to validate and repair mirrored volume copies.

Attention: Run the **repairvdiskcopy** command only if all volume copies are synchronized.

When you issue the **repairvdiskcopy** command, you must use only one of the **-validate**, **-medium**, or **-resync** parameters. You must also specify the name or ID of the volume to be validated and repaired as the last entry on the command line. After you issue the command, no output is displayed.

-validate
Use this parameter only if you want to verify that the mirrored volume copies are identical. If any difference is found, the command stops and logs an error that includes the logical block address (LBA) and the length of the first difference. You can use this parameter, starting at a different LBA each time to count the number of differences on a volume.

-medium
Use this parameter to convert sectors on all volume copies that contain different contents into virtual medium errors. Upon completion, the command logs an event, which indicates the number of differences that were found, the number that were converted into medium errors, and the number that were not converted. Use this option if you are unsure what the correct data is, and you do not want an incorrect version of the data to be used.

-resync
Use this parameter to overwrite contents from the specified primary volume copy to the other volume copy. The command corrects any differing sectors by copying the sectors from the primary copy to the copies that are being compared. Upon completion, the command process logs an event, which indicates the number of differences that were corrected. Use this action if you are sure that either the primary volume copy data is correct or that your host applications can handle incorrect data.

-startlba *lba*
Optionally, use this parameter to specify the starting Logical Block Address (LBA) from which to start the validation and repair. If you previously used the **validate** parameter, an error was logged with the LBA where the first difference, if any, was found. Reissue **repairvdiskcopy** with that LBA to avoid reprocessing the initial sectors that compared identically. Continue to reissue **repairvdiskcopy** by using this parameter to list all the differences.

Issue the following command to validate and, if necessary, automatically repair mirrored copies of the specified volume:

```
repairvdiskcopy -resync -startlba 20 vdisk8
```

Notes:

1. Only one **repairvdiskcopy** command can run on a volume at a time.
2. After you start the **repairvdiskcopy** command, you cannot use the command to stop processing.
3. The primary copy of a mirrored volume cannot be changed while the **repairvdiskcopy -resync** command is running.
4. If there is only one mirrored copy, the command returns immediately with an error.
5. If a copy that is being compared goes offline, the command is halted with an error. The command is not automatically resumed when the copy is brought back online.
6. In the case where one copy is readable but the other copy has a medium error, the command process automatically attempts to fix the medium error by writing the read data from the other copy.
7. If no differing sectors are found during **repairvdiskcopy** processing, an informational error is logged at the end of the process.

Checking the progress of validation and repair of volume copies by using the CLI

Use the **lsrepairvdiskcopyprogress** command to display the progress of mirrored volume validation and repairs. You can specify a volume copy by using the **-copy** *id* parameter. To display the volume that has two or more copies with an active task, specify the command with no parameters; it is not possible to have only one volume copy with an active task.

To check the progress of validation and repair of mirrored volumes, issue the following command:

```
lsrepairvdiskcopyprogress –delim :
```

The following example shows how the command output is displayed:

```
vdisk_id:vdisk_name:copy id:task:progress:estimated_completion_time
0:vdisk0:0:medium:50:070301120000
0:vdisk0:1:medium:50:070301120000
```

Repairing a thin-provisioned volume using the CLI

You can use the **repairsevdiskcopy** command from the command-line interface to repair the metadata on a thin-provisioned volume.

The **repairsevdiskcopy** command automatically detects and repairs corrupted metadata. The command holds the volume offline during the repair, but does not prevent the disk from being moved between I/O groups.

If a repair operation completes successfully and the volume was previously offline because of corrupted metadata, the command brings the volume back online. The only limit on the number of concurrent repair operations is the number of volume copies in the configuration.

When you issue the **repairsevdiskcopy** command, you must specify the name or ID of the volume to be repaired as the last entry on the command line. Once started, a repair operation cannot be paused or canceled; the repair can be terminated only by deleting the copy.

Attention: Use this command only to repair a thin-provisioned volume that has reported corrupt metadata.

Issue the following command to repair the metadata on a thin-provisioned volume:

```
repairsevdiskcopy vdisk8
```

After you issue the command, no output is displayed.

Notes:

1. Because the volume is offline to the host, any I/O that is submitted to the volume while it is being repaired fails.
2. When the repair operation completes successfully, the corrupted metadata error is marked as fixed.
3. If the repair operation fails, the volume is held offline and an error is logged.

Checking the progress of the repair of a thin-provisioned volume by using the CLI

Issue the **lsrepairsevdiskcopyprogress** command to list the repair progress for thin-provisioned volume copies of the specified volume. If you do not specify a volume, the command lists the repair progress for all thin-provisioned copies in the system.

Note: Run this command only after you run the **repairsevdiskcopy** command, which you must run only as required by the fix procedures that are recommended by your support team.

Recovering offline volumes using the CLI

If a node or an I/O group fails, you can use the command-line interface (CLI) to recover offline volumes.

About this task

If you lose both nodes in an I/O group, you lose access to all volumes that are associated with the I/O group. To regain access to the volumes, you must perform one of the following procedures. Depending on the failure type, you might have lost data that was cached for these volumes and the volumes are now offline.

Data loss scenario 1

One node in an I/O group failed and failover started on the second node. During the failover process, the second node in the I/O group fails before the data in the write cache is flushed to the backend. The first node is successfully repaired but its hardened data is not the most recent version that is committed to the data store, therefore, it cannot be used. The second node is repaired or replaced and lost its hardened data, and the node has no way of recognizing that it is part of the system.

Complete the following steps to recover an offline volume when one node has down-level hardened data and the other node loses hardened data.

Procedure

1. Recover the node and add it back into the system.
2. Delete all IBM FlashCopy mappings and Metro Mirror or Global Mirror relationships that use the offline volumes.
3. Run the **recovervdisk**, **recovervdiskbyiogrp** or **recovervdiskbysystem** command.
4. Re-create all FlashCopy mappings and Metro Mirror or Global Mirror relationships that use the volumes.

Example

Data loss scenario 2

Both nodes in the I/O group failed and have been repaired. Therefore, the nodes that lost their hardened data and have no way of recognizing that they are part of the system.

Complete the following steps to recover an offline volume when both nodes that have lost their hardened data and cannot be recognized by the system.

1. Delete all FlashCopy mappings and Metro Mirror or Global Mirror relationships that use the offline volumes.
2. Run the **recovervdisk**, **recovervdiskbyiogrp** or **recovervdiskbysystem** command.
3. Re-create all FlashCopy mappings and Metro Mirror or Global Mirror relationships that use the volumes.

Recovering a node and returning it to the system by using the CLI

After a node or an I/O group fails, you can use the command-line interface (CLI) to recover a node and return it to the system.

About this task

Complete the following steps to recover a node and return it to the system.

Procedure

1. Run the **lsnode** command to verify that the node is offline.
2. Run the **rmnode** *nodename_or_ID* command to remove the old instance of the offline node from the system.

3. Run the **lsnodecandidate** command to verify that the node is visible on the fabric.
4. Run the **addnode** command to add the node back into the system. In the following command, *wwnn* is the worldwide node name, *iogroupname_or_ID* identifies the I/O group, and *nodename* is the name of the node.

   ```
   addnode -wwnodename wwnn -iogrp iogroupname_or_ID -name nodename
   ```

   Note: In a service situation, a node must be added back into a system that uses the original node name. If the partner node in the I/O group has not also been deleted, this is the default name that is used if the **-name** parameter is not specified.
5. Run the **lsnode** command to verify that the node is online.

Recovering offline volumes using the CLI

You can recover offline volumes using the command-line interface (CLI).

About this task

Complete the following steps to recover offline volumes:

Procedure

1. Issue the following CLI command to list all volumes that are offline and belong to an I/O group, enter:

   ```
   lsvdisk -filtervalue IO_group_name=
    IOGRPNAME/ID:status=offline
   ```

   where *IOGRPNAME/ID* is the name of the I/O group that failed.
2. To acknowledge data loss for a volume with a *fast_write_state* of **corrupt** and bring the volume back online, enter:

   ```
   recovervdisk vdisk_id | vdisk_name
   ```

   where *vdisk_id* | *vdisk_name* is the name or ID of the volume.

   Notes:
   - If the specified volume is space-efficient or has space-efficient copies, the **recovervdisk** command starts the space-efficient repair process.
   - If the specified volume is mirrored, the **recovervdisk** command starts the resynchronization process.
3. To acknowledge data loss for all virtual disks in an I/O group with a *fast_write_state* of **corrupt** and bring them back online, enter:

   ```
   recovervdiskbyiogrp io_group_id | io_group_name
   ```

   where *io_group_id* | *io_group_name* is the name or ID of the I/O group.

   Notes:
   - If any volume is space-efficient or has space-efficient copies, the **recovervdiskbyiogrp** command starts the space-efficient repair process.
   - If any volume is mirrored, the **recovervdiskbyiogrp** command starts the resynchronization process.
4. To acknowledge data loss for all volumes in the clustered system with a *fast_write_state* of **corrupt** and bring them back online, enter:

   ```
   recovervdiskbycluster
   ```

   Notes:
   - If any volume is space-efficient or has space-efficient copies, the **recovervdiskbycluster** command starts the space-efficient repair process.
   - If any volume is mirrored, the **recovervdiskbycluster** command starts the resynchronization process.

Moving offline volumes to their original I/O group using the CLI

You can move offline volumes to their original I/O group by using the command-line interface (CLI).

About this task

After a node or an I/O group fails, you can use the following procedure to move offline volumes to their original I/O group. The system disables moving a volume if the selected volume is formatting. After the formatting completes, you can move the volume.

Attention: Do not move volumes to an offline I/O group. Ensure that the I/O group is online before you move the volume back to avoid any further data loss.

Complete the following steps to move offline volumes to their original I/O group:

Procedure

1. Enter the following command to move the volume back into the original I/O group.

   In the example, 7 is the name of the node that you want to move the volume, *IOGRP3* identifies the I/O group that you want to migrate the volume to, and *DB_volume* identifies the volume that you want to migrate.

   ```
    movevdisk -iogrp  IOGRP3 -node 7 DB_volume
   ```
2. Enter the following command, where *IO_grpname_or_ID* is the name or ID of the original I/O group, to verify that the volumes are now online.

   ```
   lsvdisk -filtervalue IO_group_name= IO_grpname_or_ID
   ```

Recording WWPN changes of replaced host HBAs

You can use the command-line interface (CLI) to record a change to a defined host object.

Before you begin

Sometimes it is necessary to replace the host-bus adapter (HBA) that connects the host to the SAN. You must inform the system of the new worldwide port names (WWPNs) that the replacement HBA contains.

Ensure that your switch is zoned correctly.

Procedure

To inform the system of a change to a defined host object, complete the following steps.

1. Enter the following CLI command to list the candidate HBA ports.

   ```
   lsfcportcandidate
   ```

   or

   ```
   lssasportcandidate
   ```

   You see a list of the HBA ports that are available for addition to host objects. One or more of these HBA ports corresponds with one or more WWPNs that belong to the new HBA port.
2. Locate the host object that corresponds with the host in which you replaced the HBA. The following CLI command lists all the defined host objects:

   ```
   lshost
   ```
3. Enter the following CLI command to list the WWPNs that are currently assigned to the host object.

   ```
   lshost hostobjectname
   ```

   Where *hostobjectname* is the name of the host object.

4. Enter the following CLI command to add the new ports to the existing host object.

   ```
   addhostport -fcwwpn one or more existing port names
    separated by : hostobjectname/ID
   ```

   Where *one or more existing port names separated by :* is the WWPNs that are currently assigned to the host object and *hostobjectname/ID* is the name or ID of the host object.
5. Enter the following CLI command to remove the old ports from the host object.

   ```
   rmhostport -fcwwpn one or more existing port names
    separated by : hostobjectname/ID
   ```

   Where one or more existing WWPNs separated by a colon (:) are the WWPNs that are currently assigned to the host object and *hostobjectname/ID* is the name or ID of the host object.

   Note: If the following conditions are met when volume protection is enabled for the system, the deletion of the specified host port fails.

   - It is the last active port on the host.
   - It is mapped to any volume that received I/O within the specified volume protection interval.

   If volume protection is enabled, and the host port being deleted is the last port for a host, which is mapped to any volume that received I/O within the defined volume protection time period, then the command fails. If multiple hosts are mapped to the same active volume, the system deletes the port if the host is offline.

Results

Any mappings that exist between the host object and the volumes are automatically applied to the new WWPNs Therefore, the host sees the volumes as the same SCSI LUNs as before.

What to do next

See the *IBM Multipath Subsystem Device Driver User's Guide* or the documentation that is provided with your multipathing driver for additional information about dynamic reconfiguration.

Expanding volumes by using the CLI

You can use the command-line interface (CLI) to expand a volume on Windows, AIX, or Linux systems.

About this task

Volumes that are mapped for FlashCopy cannot be expanded. The system disables expanding a volume if the selected volume is performing quick initialization. After the quick initialization completes, you can expand the volume.

Run Windows Update and apply all recommended updates to your system before you attempt to expand a volume that is mapped to a Windows host.

Determine the exact size of the source or master volume by issuing the following CLI command:

```
lsvdisk -bytes vdiskname
```

Where *vdiskname* is the name of the volume for which you want to determine the exact size.

Volumes can be expanded under Windows concurrently with I/O operations.

You can expand volumes for the following reasons:

- To increase the available capacity on a particular volume that is already mapped to a host.

- To increase the size of a volume so that it matches the size of the source or master volume, and so that it can be used in a FlashCopy mapping or Metro Mirror relationship.

You cannot expand the capacity of any volume in a Global Mirror with change volumes relationship or in a HyperSwap relationship.

You can expand the capacity of volumes in Metro Mirror and Global Mirror relationships that are in `consistent_synchronized` state if those volumes are using thin-provisioned or compressed copies. You cannot expand the following types of volumes:

- Volumes in HyperSwap relationships or in Global Mirror relationships that are operating in `cycling` mode.
- Volumes in relationships where a change volume is configured.
- Volumes that have a fully allocated copy.

You must expand both volumes in a relationship to maintain full operation of the system. Expand the secondary volume by the required capacity, and then expand the primary volume.

A volume that is not mapped to any hosts and does not contain any data can be expanded at any time. If the volume contains data that is in use, you can expand the volume if your host has a supported AIX or Microsoft Windows operating system.

For more information and restrictions on expanding volumes, see the software restrictions page on the following website: www.ibm.com/support

Expanding a volume that is mapped to an AIX host

The system supports expanding the size of a volume if the AIX host is using AIX version 5.2 or later.

About this task

The AIX **`chvg`** command option can be used to expand the size of a physical volume that the Logical Volume Manager (LVM) uses. The physical volume can be expanded without interruptions to the use or availability of the system. For more information, see the *AIX System Management Guide Operating System and Devices*.

Expanding a volume that is mapped to a Microsoft Windows host by using the CLI

You can use the command-line interface (CLI) to expand the size of a volume that is mapped to a Microsoft Windows host.

About this task

Complete the following steps to expand a volume that is mapped to a Windows host:

Procedure

1. Enter the following CLI command to expand the volume:

   ```
   expandvdisksize -size disk_size -unit data_unit vdisk_name/vdisk_id
   ```

   Where

   - *disk_size* is the capacity by which you want to expand the volume.
   - `b | kb | mb | gb | tb | pb` is the `data_unit` to use with the capacity.
   - *vdisk_name/vdisk_id* is the name of the volume or the ID of the volume to expand.
2. On the Windows host, start the Computer Management application and open the Disk Management window under the Storage branch.

Results

You see the volume that you expanded now has some unallocated space at the end of the disk.

You can expand dynamic disks without stopping I/O operations in most cases.

What to do next

If the Computer Management application was open before you expanded the volume, use the Computer Management application to issue a rescan command.

If the disk is a Windows basic disk, you can create a new primary or extended partition from the unallocated space.

If the disk is a Windows dynamic disk, you can use the unallocated space to create a new volume (simple, striped, mirrored) or add it to an existing volume.

Shrinking a volume using the CLI

You can reduce the size of a compressed or uncompressed volume by using the command-line interface (CLI).

About this task

Volumes can be reduced in size, if it is necessary. You can make a target or auxiliary volume the same size as the source or master volume when you create FlashCopy mappings, Metro Mirror relationships, or Global Mirror relationships. However, if the volume contains data, do not shrink the size of the disk. The system disables shrinking a volume if the selected volume is performing quick initialization. After the quick initialization completes, you can shrink the volume.

Attention:

1. It is difficult to anticipate how an operating system or file system uses the capacity in a volume. When you shrink a volume, capacity is removed from the end of the disk, whether or not that capacity is in use. Even if a volume has free capacity, do not assume that only unused capacity is removed when you shrink a volume.
2. If the volume contains data that is being used, *do not attempt under any circumstances to shrink a volume without first backing up your data*.
3. For performance reasons, some operating systems or file systems use the outer edge of the disk.
4. Do not shrink Global Mirror volumes or Global Mirror change volumes or run **recovervdisk**.

You can use the **shrinkvdisksize** command to shrink the physical capacity that is allocated to the particular volume by the specified amount. You can also shrink the virtual capacity of a thin-provisioned volume without altering the physical capacity that is assigned to the volume.

You cannot shrink the capacity of any volume in a Global Mirror with change volumes relationship or in a HyperSwap relationship.

You can shrink the capacity of volumes in Metro Mirror and Global Mirror relationships that are in `consistent_synchronized` state if those volumes are using thin-provisioned or compressed copies. You cannot shrink the following types of volumes:

- Volumes in HyperSwap relationships or in Global Mirror relationships that are operating in `cycling` mode.
- Volumes in relationships where a change volume is configured.
- Volumes that have a fully allocated copy.

You must shrink both volumes in a relationship to maintain full operation of the system. Shrink the primary volume by the required capacity, and then shrink the secondary volume.

Procedure

Complete the following steps to shrink a volume:

1. Validate that the volume is not mapped to any host objects. If the volume is mapped, data is displayed.
2. You can determine the exact capacity of the source or master volume. Issue the following command:

   ```
   lsvdisk -bytes vdisk_name
   ```

3. Shrink the volume by the required amount. Enter the following command, where *size_change* indicates the size reduction for the volume in the specified units and *vdisk_name* is the volume that you are shrinking.

   ```
   shrinkvdisksize -size size_change -unit
   b | kb | mb | gb | tb | pb vdisk_name
   ```

Migrating extents using the CLI

To improve performance, you can migrate extents using the command-line interface (CLI).

About this task

The system provides various data migration features. These features can be used to move the placement of data both *within* parent pools and *between* parent pools. These features can be used concurrently with I/O operations. You can use either of these methods to migrate data:

1. Migrating data (extents) from one MDisk to another (within the same parent pool). This method can be used to remove highly used MDisks.
2. Migrating volumes from one parent pool to another. This method can be used to remove highly used parent pools. For example, you can reduce the use of a pool of MDisks. Child pools that receive their capacity from parent pools, cannot have extents that are migrated to them.

Notes:

1. The source MDisk must not currently be the source MDisk for any other migrate extents operation.
2. The destination MDisk must not be the destination MDisk for any other migrate extents operation.

Migration commands fail if the target or source volume is offline, there is no quorum disk defined, or the defined quorum disks are unavailable. Correct the offline or quorum disk condition and reissue the command.

You can determine the use of particular MDisks by gathering input/output (I/O) statistics about nodes, MDisks, and volumes. After you collect this data, you can analyze it to determine which MDisks are used frequently. The procedure then takes you through querying and migrating extents to different locations in the same parent pool. This procedure can only be completed using the command-line interface.

If performance monitoring tools indicate that an MDisk in the pool is being overused, you can migrate data to other MDisks within the same parent pool.

Procedure

1. Determine the number of extents that are in use by each volume for the MDisk by issuing this CLI command:

   ```
   lsmdiskextent mdiskname
   ```

This command returns the number of extents that each volume is using on the MDisk. Select some of these extents to migrate within the pool.

2. Determine the other MDisks that are in the same volume.
   a. To determine the parent pool that the MDisk belongs to, issue this CLI command:

      ```
      lsmdisk mdiskname | ID
      ```

   b. List the MDisks in the pool by issuing this CLI command:

      ```
      lsmdisk -filtervalue mdisk_grp_name=mdiskgrpname
      ```

3. Select one of these MDisks as the target MDisk for the extents. You can determine how many free extents exist on an MDisk by issuing this CLI command:

   ```
   lsfreeextents mdiskname
   ```

   You can issue the **lsmdiskextent** *newmdiskname* command for each of the target MDisks to ensure that you are not just moving the over-utilization to another MDisk. Check that the volume that owns the set of extents to be moved does not already own a large set of extents on the target MDisk.

4. For each set of extents, issue this CLI command to move them to another MDisk:

   ```
   migrateexts -source mdiskname | ID -exts num_extents
   -target newmdiskname | ID -threads 4 -vdisk vdiskid
   ```

   where *num_extents* is the number of extents on the *vdiskid*. The *newmdiskname* | *ID* value is the name or ID of the MDisk to migrate this set of extents to.

   Note: The number of threads indicates the priority of the migration processing, where **1** is the lowest priority and **4** is the highest priority.

5. Repeat the previous steps for each set of extents that you are moving.
6. You can check the progress of the migration by issuing this CLI command:

   ```
   lsmigrate
   ```

Migrating volumes between pools using the CLI

You can migrate volumes between pools using the command-line interface (CLI).

About this task

You can determine the usage of particular MDisks by gathering input/output (I/O) statistics about nodes, MDisks, and volumes. After you collect this data, you can analyze it to determine which volumes or MDisks are hot. You can then migrate volumes from one storage pool to another.

Complete the following step to gather statistics about MDisks and volumes:

1. Use secure copy (**scp** command) to retrieve the dump files for analyzing. For example, issue the following command:

   ```
   scp clusterip:/dumps/iostats/v_*
   ```

   This command copies all the volume statistics files to the AIX host in the current directory.

2. Analyze the memory dumps to determine which volumes are hot. It might be helpful to also determine which MDisks are being used heavily as you can spread the data that they contain more evenly across all the MDisks in the storage pool by migrating the extents.

After you analyze the I/O statistics data, you can determine which volumes are hot. You also need to determine the storage pool that you want to move this volume to. Either create a new storage pool or determine an existing group that is not yet overly used. Check the I/O statistics files that you generated and then ensure that the MDisks or volumes in the target storage pool are used less than the MDisks or volumes in the source storage pool.

You can use data migration or volume mirroring to migrate data between storage pools. Data migration uses the command **migratevdisk**. Volume mirroring uses the commands **addvdiskcopy** and **rmvdiskcopy**.

Migrating data using migratevdisk

You can use the **migratevdisk** command to migrate data between two storage pools. When you issue the **migratevdisk** command, a check is made to ensure that the destination of the migration has enough free extents to satisfy the command. If it does, the command proceeds. The command takes some time to complete.

Notes:

- You cannot use the data migration function to move a volume between storage pools that have different extent sizes.
- Migration commands fail if the target or source volume is offline, there is no quorum disk defined, or the defined quorum disks are unavailable. Correct the offline or quorum disk condition and reissue the command.
- The system supports migrating volumes between child pools within the same parent pool or migrating a volume in a child pool to its parent pool. Migration of volumes fails if source and target child pools have different parent pools. However, you can use **addvdiskcopy** and **rmvdiskcopy** commands to migrate volumes between child pools in different parent pools.

When you use data migration, it is possible for the free destination extents to be consumed by another process; for example, if a new volume is created in the destination parent pool or if more migration commands are started. In this scenario, after all the destination extents are allocated, the migration commands suspend and an error is logged (error ID 020005). To recover from this situation, use either of the following methods:

- Add more MDisks to the target parent pool, which provides more extents in the group and allows the migrations to be restarted. You must mark the error as fixed before you reattempt the migration.
- Migrate one or more volumes that are already created from the parent pool to another group. This action frees up extents in the group and allows the original migrations to be restarted.

Complete the following steps to use the **migratevdisk** command to migrate volumes between storage pools:

1. After you determine the volume that you want to migrate and the new storage pool that you want to migrate it to, issue the following CLI command:
   ```
   migratevdisk -vdisk vdisk_name
    -mdiskgrp
    mdisk_group_name -threads 4
   ```
2. You can check the progress of the migration by issuing the following CLI command:
   ```
   lsmigrate
   ```

Migrating data using volume mirroring

When you use data migration, the volume goes offline if either pool fails. Volume mirroring can be used to minimize the impact to the volume because the volume goes offline only if the source pool fails. You can migrate volumes between child pools or from a child pool to a parent pool using the **addvdiskcopy** and **rmvdiskcopy** commands instead of using the **migratevdisk** command.Complete the following steps to use volume mirroring to migrate volumes between pools:

1. After you determine the volume that you want to migrate and the new pool that you want to migrate it to, enter the following command:

```
addvdiskcopy -mdiskgrp mdisk_group_name -autodelete vdisk_name
```

where *mdisk_group_name* is the name of the new storage pool and *vdisk_name* is the name of the volume that is being copied. Specify **-autodelete** to automatically delete the original copy of the volume after the copies are synchronized.

2. The copy ID of the new copy is returned. The copies now synchronize such that the data is stored in both storage pools. You can check the progress of the synchronization by issuing the following command:

   ```
   lsvdisksyncprogress
   ```

Moving a volume between I/O groups using the CLI

To move volumes between I/O groups non-disruptively, ensure that hosts are mapped to the volume, support non-disruptive volume move. The cached data that is held within the system must first be written to the system disk before the allocation of the volume can be changed.

About this task

Modifying the I/O group that services the volume can be done concurrently with I/O operations if the host supports non- disruptive volume move. It also requires a rescan at the host level to ensure that the multipathing driver is notified that the allocation of the preferred node has changed and the ports by which the volume is accessed has changed. This can be done in the situation where one pair of nodes becomes over used.

If there are any host mappings for the volume, the hosts must be members of the target I/O group or the migration fails.

Verify that you created paths to I/O groups on the host system. After the system successfully adds the new I/O group to the volume's access set and you moved selected volumes to another I/O group, detect the new paths to the volumes on the host. The commands and actions on the host vary depending on the type of host and the connection method used. These steps must be completed on all hosts to which the selected volumes are currently mapped.

You can also use the management GUI to move volumes between I/O groups non-disruptively. In the management GUI, select **Volumes** > **Volumes**. On the Volumes panel, select the volume that you want to move and select **Actions** > **Move to Another I/O Group**. The wizard guides you through all the steps that are necessary for moving a volume to another I/O group, including any changes to hosts that are required. Click **Need Help** on the associated management GUI panels for details.

Note: If the selected volume is performing quick initialization, this wizard is unavailable until quick initialization is complete.

To move a volume between I/O groups using the CLI, complete the following steps:

Procedure

1. Issue the following command: **`addvdiskaccess -iogrp iogrp id/name volume id/name`**
2. Issue the following command: **`movevdisk -iogrp destination iogrp -node new preferred node volume id/name`** The system disables moving a volume if the selected volume is currently performing quick initialization. After the quick initialization completes, you can move the volume to another I/O group.
3. Issue the appropriate commands on the hosts that are mapped to the volume to detect the new paths to the volume in the destination I/O group.
4. After confirming that the new paths are online, remove access from the old I/O group: **`rmvdiskaccess -iogrp iogrp id/name volume id/name`**

5. Issue the appropriate commands on the hosts that are mapped to the volume to remove the paths to the old I/O group.

Creating an image-mode volume using the CLI

You can use the command-line interface (CLI) to import storage that contains existing data and continue to use this storage. You can also use the advanced functions, such as Copy Services, data migration, and the cache. These disks are known as image-mode volumes.

About this task

Make sure you are aware of the following information before you create image-mode volumes:

1. Unmanaged-mode managed disks (MDisks) that contain existing data cannot be differentiated from unmanaged-mode MDisks that are blank. Therefore, it is vital that you control the introduction of these MDisks to the clustered system by adding these disks one at a time. For example, map a single LUN from your RAID storage system to the clustered system and refresh the view of MDisks. The newly detected MDisk is displayed.
2. Do not manually add an unmanaged-mode MDisk that contains existing data to a parent pool. If you do, the data is lost. When you use the command to create an image-mode volume from an unmanaged-mode disk, select the parent pool where it should be added. Ensure that the pool that is selected is not a child pool. Child pools are created from existing pools, called parent pools, and get capacity from the parent pool, not MDisks.

Complete the following steps to create an image-mode volume:

Procedure

1. Stop all I/O operations from the hosts. Unmap the logical disks that contain the data from the hosts.
2. Create one or more storage pools. Ensure that the pool is not a child pool.
3. Map a single array or logical unit from your RAID storage system to the clustered system. You can do this through a switch zoning or a RAID storage system based on your host mappings. The array or logical unit appears as an unmanaged-mode MDisk to the system.
4. Issue the **lsmdisk** command to list the unmanaged-mode MDisks.

   If the new unmanaged-mode MDisk is not listed, you can complete a fabric-level discovery. Issue the **detectmdisk** command to scan the Fibre Channel network for the unmanaged-mode MDisks.

   Note: The **detectmdisk** command also rebalances MDisk access across the available storage system device ports.
5. Convert the unmanaged-mode MDisk to an image-mode volume.

   Note: If the volume that you are converting maps to a flash drive, the data that is stored on the volume is not protected against Flash drive failures or node failures. To avoid data loss, add a volume copy that maps to an Flash drive on another node.
   Issue the **mkvdisk** command to create an image-mode volume object.
6. Map the new volume to the hosts that were previously using the data that the MDisk now contains. You can use the **mkvdiskhostmap** command to create a new mapping between a volume and a host. This makes the image-mode volume accessible for I/O operations to the host.

Results

After the volume is mapped to a host object, the volume is detected as a disk drive with which the host can complete I/O operations.

What to do next

If you want to virtualize the storage on an image-mode volume, you can transform it into a striped volume. Migrate the data on the image-mode volume to managed-mode disks in another storage pool. Issue the **migratevdisk** command to migrate an entire image-mode volume from one storage pool to another storage pool. Ensure that the storage pool that you migrate the image-mode volume to is not a child pool.

Migrating data to an image mode volume using the CLI

You can use the command-line interface (CLI) to migrate data to an image mode volume.

About this task

Use the **migratetoimage** CLI command to migrate the data from an existing volume onto a different managed disk (MDisk).

When the **migratetoimage** CLI command is issued, it migrates the data of the user specified source volume onto the specified target MDisk. When the command completes, the volume is classified as an image mode volume.

Note: Migration commands fail for the following reasons:
- The target or source volume is offline.
- A quorum disk is not defined.
- Defined quorum disks are unavailable.

Correct the offline or quorum disk condition and reissue the command.

The MDisk specified as the target must be in an unmanaged state at the time the command is run. Using this command results in the inclusion of the MDisk into the user specified storage pool.

Enter the following CLI command to migrate data to an image mode volume:

```
migratetoimage -vdisk source_vdisk_name -mdisk unmanaged_target_mdisk_name -mdiskgrp managed_disk_group_name
```

where *source_vdisk_name* is the name of the image mode volume, *unmanaged_target_mdisk_name* is the name of the new MDisk, and *managed_disk_group_name* is the name of the new storage pool. For example, the following command migrates data from the vdisk0 image mode volume to the mdisk5 target in the mdgrp2 storage pool:

```
migratetoimage -vdisk vdisk0 -mdisk mdisk5 -mdiskgrp mdgrp2
```

Deleting a node from a system by using the CLI

You can use the command line interface (CLI) to remove a node from a system.

Before you begin

After the node is deleted, the other node in the I/O group enters write-through mode until another node is added back into the I/O group.

By default, the **rmnode** command flushes the cache on the specified node before the node is taken offline. When the system is operating in a degraded state, the system ensures that data loss does not occur as a result of deleting the only node with the cache data.

Attention:

- If you are removing a single node and the remaining node in the I/O group is online, the data can be exposed to a single point of failure if the remaining node fails.
- If both nodes in the I/O group are online and the volumes are already degraded before you delete the node, redundancy to the volumes is already degraded. If the **force** option is used, removing a node might result in loss of access to data, and data loss might occur.
- Removing the last node destroys the system. Before you delete the last node in the system, ensure that you want to destroy the system.
- When you delete a node, you remove all redundancy from the I/O group. As a result, new or existing failures can cause I/O errors on the hosts. The following failures can occur:
  - Host configuration errors
  - Zoning errors
  - Multipathing software configuration errors
- If you are deleting the last node in an I/O group and volumes are assigned to the I/O group, you cannot delete the node from the system if the node is online. You must back up or migrate all data that you want to save before you delete the node. If the node is offline, you can delete the node.
- To take the specified node offline immediately without flushing the cache or ensuring that data loss does not occur, run the **rmnode** command with the **force** parameter. The **force** parameter forces continuation of the command even though any node-dependent volumes will be taken offline. Use the **force** parameter with caution; access to data on node-dependent volumes will be lost.
- To delete a node that is in the service state and that has an associated spare node, you must specify the **-deactivatespare** parameter with the **rmnode** command.

About this task

Complete these steps to delete a node:

Procedure

1. If you are deleting the last node in an I/O group, determine the volumes that are still assigned to this I/O group:
   a. Issue the following CLI command to request a filtered view of the volumes:
      ```
      lsvdisk -filtervalue IO_group_name=name
      ```
      Where *name* is the name of the I/O group.
   b. Issue the following CLI command to list the hosts that this volume is mapped to:
      ```
      lsvdiskhostmap vdiskname/identification
      ```
      Where *vdiskname/identification* is the name or identification of the volume.

   Note: If volumes are assigned to this I/O group that contain data that you want to continue to access, back up the data or migrate the volumes to a different (online) I/O group.
2. Turn off the power to the node that you intend to remove, if this node is not the last node in the clustered system. This step ensures that the multipathing device driver, such as the Subsystem Device Driver (SDD), does not rediscover the paths that are manually removed before you issue the delete node request.

Attention:

a. If you are removing the configuration node, the **rmnode** command causes the configuration node to move to a different node within the clustered system. This process might take a short time, typically less than a minute. The system IP address remains unchanged, but any SSH client that is attached to the configuration node must reestablish a connection.

b. If you turn on the power to the node that was removed and it is still connected to the same fabric or zone, it attempts to rejoin the system. The system causes the node to remove itself from the system and the node becomes a candidate to add to this system or another system.

c. If you are adding this node into the system, ensure that you add it to the same I/O group that it was previously a member of. Failure to do so can result in data corruption.

d. In a service situation, a node should normally be added back into a system using the original node name. As long as the partner node in the I/O group was not deleted too, this is the default name used if **-name** is not specified.

3. Before you delete the node, update the multipathing device driver configuration on the host to remove all device identifiers that are presented by the volumes that you intend to remove. If you are using the Subsystem Device Driver, the device identifiers are referred to as virtual paths (vpaths).

   Attention: Failure to complete this step can result in data corruption.

   See the *IBM Multipath Subsystem Device Driver User's Guide* for details about how to dynamically reconfigure SDD for the host operating system.

4. Issue this CLI command to delete a node from the clustered system:

   Attention: Before you delete the node, note the following information. The **rmnode** command checks for node-dependent volumes, which are not mirrored at the time that the command is run. If any node-dependent volumes are found, the command stops and returns a message. To continue removing the node despite the potential loss of data, run the **rmnode** command with the **force** parameter. Alternatively, follow these steps before you remove the node to ensure that all volumes are mirrored:

   a. Run the **lsdependentvdisks** command.

   b. For each node-dependent volume that is returned, run the **lsvdisk** command.

   c. Ensure that each volume returns in-sync status.

   ```
   rmnode node_name_or_identification
   ```

   Where *node_name_or_identification* is the name or identification of the node.

   Note: Before a node is removed, the command checks for any node-dependent volumes that would go offline. If the node that you selected to delete contains a flash drive that has dependent volumes, volumes that use the flash drives go offline and become unavailable if the node is deleted. To maintain access to volume data, mirror these volumes before you remove the node. To continue removing the node without mirroring the volumes, specify the **force** parameter.

Completing the system maintenance procedure by using the CLI

You can use the command-line interface (CLI) to complete the system maintenance procedure.

About this task

Use the following steps to complete the system maintenance procedure:

Procedure

1. Issue the **finderr** command to analyze the error log for the highest severity of unfixed errors. This command scans the error log for any unfixed errors. Given a priority order that is defined within the code, the highest priority of unfixed errors is returned.
2. Issue the **dumperrlog** command to dump the contents of the error log to a text file.
3. Locate and fix the error.

4. Issue the **clearerrlog** command to clear all entries from the error log, including status events and any unfixed errors. Only issue this command when you rebuild the system or you fix a major problem that caused many entries in the error log that you do not want to fix individually.

   Note: Clearing the error log does not fix the errors.
5. Issue the **cherrstate** command to toggle the state of an error between `unfixed` and `fixed`.

Modifying system IP addresses using the CLI

Use the command-line interface (CLI) to change the IP addresses that are associated with a system.

About this task

Attention: When you specify a new IP address for a system, the existing communication with the system is broken. You must reconnect to the system with the new IP address. Additionally, the address for a system IP cannot be the same address that is used for the service IP. Using the same IP address causes an error.

Procedure

To change the system IP address, complete the following steps:

1. Issue the **lssystemip** command to list IP addresses that are used by the system.
2. Record the IP addresses for future reference.
3. To change an Internet Protocol Version 4 (IPv4) system IP address, issue this command:
   ```
   chsystemip -clusterip cluster_ip_address -port cluster_port
   ```
   where *cluster_ip_address* is the new IP address for the system and *cluster_port* specifies the port (1 or 2) where changes apply.
4. To change an IPv4 system IP address to an IPv6 system IP address, issue this command:
   ```
   chsystemip -clusterip_6 cluster_ip_address -port cluster_port
   ```
   where *cluster_ip_address* is the new Internet Protocol Version 6 (IPv6) address for the system and *cluster_port* specifies the port (1 or 2) where changes apply.
5. To change an IPv4 default gateway IP address, issue this command:
   ```
   chsystemip -gw cluster_gateway_address -port cluster_port
   ```
   where *cluster_gateway_address* is the new gateway address for the system and *cluster_port* specifies the port (1 or 2) where changes apply.
6. To change an IPv6 default gateway address, issue this command:
   ```
   chsystemip -gw_6 cluster_gateway_address -port cluster_port
   ```
   where *cluster_gateway_address* is the new gateway address for the system and *cluster_port* specifies the port (1 or 2) where changes apply.
7. Issue this command to change an IPv4 system subnet mask
   ```
   chsystemip -mask cluster_subnet_mask -port cluster_port
   ```
   where *cluster_subnet_mask* is the new subnet mask for the system and *cluster_port* specifies the port (1 or 2) where changes apply.
8. For IPv6 addresses, you can issue this command to set the prefix for the system:
   ```
   chsystemip -prefix_6 -port cluster_port
   ```
   where *cluster_port* specifies the port (1 or 2) where changes apply.
9. Optionally, to delete all of the IPv4 addresses in the system after you change all addresses to IPv6, issue this command:
   ```
   chsystem -noip
   ```

10. Optionally, to delete all of the IPv6 addresses in the system after you change all addresses to IPv4, issue this command:

    ```
    chsystem -noip_6
    ```

11. Display the IP routing table (optional) by using the CLI command `lsroute` as follows:

    ```
    lsroute
    ```

    The IP routing table provides details of the gateway that is used for IP traffic to a range of IP addresses for each Ethernet port. This information can be used to diagnose configuration node accessibility problems.

12. Issue the **ping** command (optional) to diagnose IP configuration problems. An example of the command is as follows:

    ```
    ping -srcip4 source_ipv4_address destination_ipv4_address -srcip6 source_ipv6_address destination_ipv6_address
    ```

Changing the system gateway address by using the CLI

You can use the command-line interface (CLI) to change the gateway address for a system.

Procedure

To change the system gateway address, complete the following steps:

1. Issue the **lssystemip** command to list the current gateway address of the system.
2. Record the current gateway address for future reference.
3. Issue the following command to change an IPv4 system gateway address:

   ```
   chsystemip -gw cluster_gateway_address -port cluster_port
   ```

   where *cluster_gateway_address* is the new gateway address for the system. The **port** parameter specifies which port (1 or 2) to apply changes to.

4. Issue the following command to change an IPv6 system gateway address:

   ```
   chsystemip -gw_6 cluster_gateway_address -port cluster_port
   ```

   where *cluster_gateway_address* is the new gateway address for the system. The **port** parameter specifies which port (1 or 2) to apply changes to.

Changing the relationship bandwidth for a system by using the CLI

You can use the command-line interface (CLI) to change the relationship bandwidth for a system.

About this task

The relationship bandwidth limit controls the maximum rate at which any one remote-copy relationship can synchronize. The overall limit is controlled by the **bandwidth** parameter of each system partnership. The default value for the relationship bandwidth limit is 25 megabytes per second (MBps), but you can change this value by following these steps:

Procedure

1. Issue the **lssystem** command to list the current relationship bandwidth limit of the system. For example:

   ```
   lssystem system_id_or_system_name
   ```

   Where *system_id_or_system_name* is the ID or name of the system.

2. For future reference, record the current relationship bandwidth limit that is displayed. For example:

   ```
   relationship_bandwidth_limit 25
   ```

3. To change the relationship bandwidth limit of the system, issue the following command:

```
chsystem -relationshipbandwidthlimit
 system_relationship_bandwidth_limit
```

Where *system_relationship_bandwidth_limit* is the new limit for the system. If the system is in a remote partnership with another system, then the bandwidth setting must be the same between both systems in the partnership. Issue the command on both systems in the partnership. Issue the command on both systems in a relationship.

Configuring the system for iSCSI hosts

You need to complete several tasks to configure the system to work with iSCSI-attached hosts. The tasks include general tasks on the host system before you configure a system.

Before you begin

Before you complete any iSCSI-configuration tasks on the system, it is important that you complete all the iSCSI-related configuration on the host machine. Because the system supports various host machines, consult the documentation for specific instructions and requirements for a particular host. For a list of supported hosts, see this website:

www.ibm.com/support

About this task

To configure a system for iSCSI, follow these general tasks on the host system:

1. Select a software-based iSCSI initiator, such as Microsoft Windows iSCSI Software Initiator and verify the iSCSI driver installation.
2. If required, install and configure a multipathing driver for the host system.

In addition, determine a naming convention for iSCSI names, such as iSCSI qualified names (IQNs) for your system. Hosts use iSCSI names to connect to the node. Each node, for example, has a unique IQN, and the system name and node name are used as part of that IQN. Each node, for example, has a unique IQN, and the system name and node name are used as part of that IQN.

Port IP addresses are the IP addresses that are used by iSCSI-attached hosts to process I/O. Host port group IDs are automatically assigned to the port. Host port grouping groups the ports that have the same maximum possible port speed and ensures that no more than four ports are discovered by a host.

Procedure

1. To configure a new port IP address to a specified Ethernet port of a node with an IPv4 address, enter the following command-line interface (CLI) command:

   ```
   cfgportip -node  -ip ipv4addr
   -gw ipv4gw -mask subnet_mask -failover -vlan vlan_id port_id
   ```

   where *node_name* | *node_id* specifies the name or ID of the node that is being configured, *ipv4addr* is the IPv4 address for the Ethernet port, *ipv4gw* is the IPv4 gateway IP address, *subnet_mask* is the IPv4 subnet mask, and *port_id* specifies the Ethernet port ID (1 or 2). To view a list of ports, use the **lsportip** command.

   The optional **-failover** parameter specifies that the IP is a failover IP and is related to the partner node. If the node that is specified is the only online node in the I/O group, the address is configured and presented by this node. When another node in the I/O group comes online, the failover address is presented by that node. If two nodes in the I/O group are online when the command is entered, the address is presented by the other node to the partner node.

   The optional **-vlan** parameter sets the virtual local area network (VLAN) ID for an IPv4 address that is configured for iSCSI host attachment.

For more information about the **-vlan** parameter, see the information about configuring VLAN for iSCSI using the CLI.

2. To configure a new port IP address to a specified Ethernet port of a node with an IPv6 address, enter the following CLI command:

   ```
   cfgportip -node node_name | node_id -ip_6 ipv6addr
   -gw_6 ipv6gw -prefix_6 prefix -failover -vlan_6 vlan_id port_id
   ```

   where *node_name* | *node_id* specifies the name or ID of the node that is being configured, *ipv6addr* is the IPv6 address for the Ethernet port, *ipv6gw* is the IPv6 gateway IP address, *subnet_mask* is the IPv6 subnet mask, and *port_id* specifies the Ethernet port ID (1 or 2). To view a list of ports, use the **lsportip** command.

   The optional **-failover** parameter specifies that the IP is a failover IP that is related to the partner node. If the node that is specified is the only online node in the I/O group, the address is configured and presented by this node. When another node in the I/O group comes online, the failover address is presented by that node. If two nodes in the I/O group are online when the command is entered, the address is presented by the other node to the node that is specified.

   The optional **-vlan_6** parameter sets the virtual local area network (VLAN) ID for an IPv6 address that is configured for iSCSI host attachment.

   For more information about the **-vlan** parameter, see the information about configuring VLAN for iSCSI using the CLI.

3. After IP configuration, host_port_group_ids are automatically assigned to the iSCSI ports and consist of the following criteria:
   - A host_port_group_id is an automatic grouping of ports that is designated by an integer. Host port group IDs are unique across I/O groups.
   - Each host port group ID contains a maximum of four ports.
   - All ports within a host port group ID have identical speeds
   - Identical host port group IDs are assigned to the failover port. If a host_port_group_id is already assigned to a failover port, the same host_port_group_id will be assigned to a local port
   - Enabling -host flag to *yes* assigns the host_port_group_id. If on a port with host flag *no* , host flag is set to *yes*, this results in assignment of a host_port_group_id to a port.

4. To remove an iSCSI IP address from a node Ethernet port, enter either of these CLI commands. The following command deletes an IPv4 configuration for the specified iSCSI Ethernet port:

   ```
   rmportip -failover
   -node node_name | node_id port_id
   ```

   where *node_name* | *node_id* specifies the name or ID of the node with the Ethernet port that the IP address is being removed from and *port_id* specifies the Ethernet port ID. To list the valid values for the Ethernet port, enter the **lsportip** command. The optional **-failover** parameter indicates that the specified data is failover data.

   The following command deletes an IPv6 configuration for the specified iSCSI Ethernet port:

   ```
   rmportip -ip_6 -failover
   -node node_name | node_id port_id
   ```

   where **-ip_6** indicates that this command removes an IPv6 configuration, *node_name* | *node_id* specifies the name or ID of the node with the Ethernet port that the IP address is being removed from, and *port_id* specifies the Ethernet port ID. To list the valid values for the Ethernet port, enter the **lsportip** command. The optional **-failover** parameter indicates that the specified data is failover data.

5. To display the host port group ID in addition to other parameters for each iSCSI port, enter the **lsportip** command. Entering this command displays a detailed view of the specified port:

   ```
   lsportip Ethernet_port_id
   ```

   where *Ethernet_port_id* is the specified port. The parameter **host_port_grp_id** displays the value of the host port group ID.

6. After all the IP addresses of the ports are removed, the host port group ID that is associated with a port is removed. The host port group ID is also removed when the **-host** flag is set to *no* from *yes* on the port.

What to do next

After you configure your IP addresses, you can optionally create or configure several iSCSI items.

Configuring or modifying an iSCSI alias by using the CLI

You can use the command-line interface (CLI) to optionally create or change the iSCSI alias for the selected node. An iSCSI alias is a user-assigned name that identifies the node to the iSCSI-attached host.

About this task

To configure or modify an iSCSI alias, follow these steps:

Procedure

1. To configure a new port IP address to a specified Ethernet port of a node, enter the following CLI command:

   ```
   chnode -iscsialias  alias node_name | node_id
   ```

   where *alias node_name* | *node_id* specifies the name or ID of the node.
2. To specify that the name or iSCSI alias that is being set is the name or alias of the partner node in the I/O group, enter the following CLI command. When there is no partner node, the values set are applied to the partner node when it is added to the clustered system. If this parameter is used when there is a partner node, the name or alias of that node changes

   ```
   chnode -iscsialias alias -failover node_name | node_id
   ```

   where *alias* specifies the iSCSI name of the node and *node_name* | *node_id* specifies the node to be modified.

What to do next

After you create iSCSI aliases, you can optionally configure the address for the Internet Storage Name Service (iSNS) server for the system.

Configuring the iSNS server address by using the CLI

If you are using iSCSI-attached hosts with the clustered system, you can use the command-line interface (CLI) to optionally configure the address for the internet Storage Name Service (iSNS) server for the system. Host systems use the iSNS server to manage iSCSI targets and for iSCSI discovery.

Procedure

1. To specify an IPv4 address for the iSCSI storage name service (SNS), enter the following CLI command:

   ```
   chsystem -isnsip sns_server_address
   ```

   where *sns_server_address* is the IP address of the iSCSI storage name service in IPv4 format.
2. To specify an IPv6 address for the iSCSI storage name service (SNS), enter the following CLI command:

   ```
   chsystem -isnsip_6 ipv6_sns_server_address
   ```

   where *ipv6_sns_server_address* is the IP address of the iSCSI storage name service in IPv6 format.

Configuring system iSCSI authentication by using the CLI

You can use the command-line interface (CLI) to configure the system to authenticate with iSCSI-attached hosts by using the Challenge-Handshake Authentication Protocol (CHAP). After the CHAP is set for the system, all attached hosts must be configured to authenticate. When you are troubleshooting a problem, you can delay your configuration of the CHAP authentication until after you configure the first one or two hosts and test their connectivity.

About this task

To configure authentication between the system and the iSCSI-attached hosts, follow these steps:

Procedure

1. To configure CHAP authentication for an iSCSI host, enter the following CLI command:

   ```
   chhost -iscsiusername iscsi_username -chapsecret chap_secret host_name
   ```

   where *iscsi_username* is the user name, *chap_secret* is the CHAP secret to be used to authenticate the system via iSCSI, and *host_name* is the name of the iSCSI host. The *chap_secret* value must be 12 characters. If you do not specify the iSCSI user name, the initiator's IQN is taken as the user name for one-way CHAP authentication.
2. To set the authentication method for the iSCSI communications of the system, enter the following CLI command:

   ```
   chsystem -iscsiauthmethod chap -chapsecret chap_secret
   ```

   where *chap* specifies that CHAP is the authentication method and *chap_secret* is the CHAP secret to be used. The specified CHAP secret cannot begin or end with a space.
3. To clear all CHAP secrets for iSCSI authentication that were previously set, enter the following CLI command:

   ```
   chsystem -nochapsecret
   ```

   The **nochapsecret** parameter is not allowed if the **chapsecret** parameter is specified.
4. Run the **lsiscsiauth** command to display the Challenge Handshake Authentication Protocol (CHAP) secret that you configured.

What to do next

After you configure the CHAP secret for the system, ensure that the system CHAP secret is added to each iSCSI-attached host. On all iSCSI-attached hosts, specify a CHAP secret that the hosts use to authenticate to the system.

Configuring remote authentication service using the CLI

Remote authentication allows users to authenticate to the system using credentials stored on an external authentication service.

About this task

When you configure remote authentication, you do not need to configure users on the system or assign additional passwords. You can use your existing passwords and user groups that are defined on the remote service to simplify user management and access, to enforce password policies more efficiently, and to separate user management from storage management.

If a user is configured on the system as a local user, only local credentials are used. Otherwise, users who are entering their password are authenticated against the remote service when they use the management GUI or the command-line interface (CLI). Their roles are determined according to group memberships defined on the remote service. If a user is configured on the system as a remote user with an SSH key, the user can additionally access the command-line interface by using this Secure Shell (SSH) key. Group memberships continue to be determined from the remote service.

Configuring remote authentication service with Lightweight Directory Access Protocol (LDAP) by using the CLI

You can use the command-line interface (CLI) to configure the system to authenticate users against servers that implement the Lightweight Directory Access Protocol (LDAP), including Active Directory (AD).

About this task

- Users on provisioned LDAP servers with IBMRBS permissions of Supervisor Access or Supervisor Role can log in to the system as Administrator, but cannot run the **satask changelocale** command.
- All authentication commands and settings are disabled.
  - Automatically provisioned settings are not visible to the user and are not displayed by the **lssystem** or **lsldapserver** commands.
  - The **chauthservice -refresh** command is enabled.

All options on the system GUI LDAP page are disabled.

Tip: A superuser cannot be authenticated if the superuser is using a remote Lightweight Directory Access Protocol (LDAP server). However, other users can authenticate in this manner.

Procedure

To enable user authentication with LDAP, follow these steps:

1. Configure LDAP by entering the **chldap** command.

   This command provides default settings for both Tivoli® Directory Server and AD. To configure authentication with Tivoli Directory Server schema defaults and Transport Layer Security (TLS), for example, enter the following command:

   ```
   chldap -type itds -security tls
   ```

   LDAP configuration can be inspected with the **lsldap** command.

   Note: Use TLS so that transmitted passwords are encrypted.
2. Specify the **mkldapserver** command to define up to six LDAP servers to use for authentication.

   Multiple servers can be configured to provide access to different sets of users or for redundancy. All servers must share the settings that are configured with **chldap**. To configure an LDAP server with a Secure Socket Layer (SSL) certificate and users in the cn=users,dc=company,dc=com subtree, for example, enter the following command:

   ```
   mkldapserver -ip 9.71.45.108 -basedn cn=users,dc=company,dc=com -sslcert /tmp/sslcert.pem
   ```

   You can also configure which servers are preferred to authenticate users.

   Specify **lsldapserver** for LDAP server configuration information. Specify **chldapserver** and **rmldapserver** to change the configured LDAP servers.
3. Configure user groups on the system by matching those user groups that are used by the authentication service.

   For each group of interest that is known to the authentication service, a system user group must be created with the same name and with the remote setting enabled. If members of a group that is called sysadmins, for example, require the system administrator (admin) role, enter the following command:

   ```
   mkusergrp -name sysadmins -remote -role Administrator
   ```

   If none of the user groups match a system user group, the user cannot access the system.
4. Verify your LDAP configuration by using the **testldapserver** command.

To test the connection to the LDAP servers, enter the command without any options. A user name can be supplied with or without a password to test for configuration errors. To process a full authentication attempt against each server, enter the following commands:

```
testldapserver -username username -password password
```

5. Enter the following command to enable LDAP authentication:

   ```
   chauthservice -type ldap -enable yes
   ```

6. Configure users who do not require Secure Shell (SSH) key access.

   Delete system users who must use the remote authentication service and do not require SSH key access.

   Remember: A superuser cannot be deleted or use the remote authentication service.

7. Configure users who require SSH key access.

   All system users who use the remote authentication service and require SSH key access must have remote settings that are enabled and a valid SSH key that is configured on the system.

Changing user groups

You can use the command-line interface (CLI) to change user groups. User groups organize users of a clustered system by role.

About this task

Roles apply to both local and remote users on the system and are based on the user group to which the user belongs. A local user can belong only to a single group; therefore, the role of a local user is defined by the single group that the user belongs to. Remote users can belong to one or more groups; therefore, the roles of remote users are assigned according to the groups that the remote user belongs to.

To change a user group in the management GUI, select **Access** > **Users**. Select a user group and select **Properties** from the **Actions** menu.

Procedure

1. Use the **chusergrp** CLI command to change attributes of an existing user group. For example, enter the following command:

   ```
   chusergrp -role role_name -remote yes | no group_id_or_name
   ```

   where *role_name* specifies the role that is associated with any users that belong to this group and *group_id_or_name* specifies the group to be changed. The **remote** parameter specifies whether the group is visible to the authentication server.

2. Issue the **lsusergrp** CLI command to display the user groups that were created on the system. For example, enter the following command:

   ```
   lsusergrp usergrp_id_or_name
   ```

   where *group_id_or_name* specifies the user group to view. If you do not specify a user group ID or name, all user groups on the system are displayed.

Changing users

You can use the command-line interface (CLI) or the management GUI to change users on the system.

Before you begin

System users must provide either a password, a Secure Shell (SSH) key, or both. Local users are authenticated through the authentication methods that are on the system.

You can create two categories of users that access the clustered system (system). These user types are based on how they authenticate to the system:

- Some users must provide an SSH password (or if not possible an SSH key).
- If a user needs access to the management GUI, a password is needed for the user.
- If the user requires access to the command-line interface (CLI), a valid password and SSH key can be used.
- Users must be in a user group that is defined on the system.

Remote users can also configure local credentials if they need to access the system when the remote service is down. Remote users have their groups that are defined by the remote authentication service.

To change a user in the management GUI, select **Access** > **Users**. Right-click the user and select **Modify** from the **Actions** menu.

About this task

To change a user in the CLI, follow these steps:

Procedure

1. Use the **chuser** CLI command to change the attributes of an existing user. For example, enter the following command:

   ```
   chuser -usergrp group_id_or_name user_id_or_name
   ```

   where the *group_id_or_name* specifies the new group for the user and *user_id_or_name* specifies the user to be changed.
2. Use the **chcurrentuser** CLI command to change the attributes of the current user. For example, enter the following command:

   ```
   chcurrentuser -nokey
   ```

   where the **nokey** parameter specifies that the SSH key of the user is to be deleted.
3. Use the **lscurrentuser** CLI command to display the name and role of the logged-in user. For example, enter the following command:

   ```
   lscurrentuser
   ```

   The name and the role of the user are displayed.

Managing SNMP notifications by using the CLI

You can set up and manage event and call home notifications by using the command-line interface (CLI).

About this task

The notification settings apply to the entire system. You can specify the types of events that cause the system to send a notification. The system sends a Simple Network Management Protocol (SNMP) notification. The SNMP setting represents the type of notification.

SNMP is the standard protocol for managing networks and exchanging messages. SNMP enables your system to send external messages that notify personnel about an event. You can use an SNMP manager to view the messages that the SNMP agent sends.

The possible types of event notifications are error, warning, and information. Event notifications are reported to the SNMP destinations of your choice. To specify an SNMP destination, you must provide a valid IP address and SNMP community string.

Note: A valid community string can contain up to 60 letters or digits. If no community string is specified, the default string of **public** is used. A maximum of six SNMP destinations can be specified.

In configurations that use SNMP, the system uses the notifications settings to call home if errors occur. You must specify Error and send the trap to the master console if you want the system to call home when errors occur.

To configure the SNMP notification settings, use the following commands:

Procedure

1. To create a new SNMP server to receive notifications, use the **mksnmpserver** CLI command. For example, enter one of the following commands:

   ```
   mksnmpserver -ip 9.11.255.634
   ```

   where *9.11.255.634* is the IP address for this server.

   ```
   mksnmpserver -ip 9.11.255.634 -port  remoteportnumber
   ```

   where *9.11.255.634* is the IP address for this server and *remoteportnumber* is the port number for the remote SNMP server.
2. To change the settings of an existing SNMP server, enter the **chsnmpserver** command. For example, enter the following command:

   ```
   chsnmpserver -name server_name snmp_server_name_or_id
   ```

   where *server_name* is the new name of the server and *snmp_server_name* is the name or ID of the server to be modified.
3. To remove an existing SNMP server from the system, enter the **rmsnmpserver** command. For example, enter the following command:

   ```
   rmsnmpserver snmp_server_name
   ```

   where *snmp_server_name* is either the name of the SNMP server to be deleted.
4. To display either a concise list or a detailed view of the SNMP servers that are detected by the system, enter the **lssnmpserver** command. For example, to display a concise view, enter the following command:

   ```
   lssnmpserver -delim :
   ```

   To display a detailed view of an SNMP server, enter the following command:

   ```
   lssnmpserver snmp_server_name
   ```

Setting up syslog notifications using the CLI

You can set up syslog event notifications by using the command-line interface (CLI).

About this task

The syslog protocol is a standard protocol for forwarding log messages from a sender to a receiver on an IP network. The system can send syslog messages that notify personnel about an event. The system can transmit syslog messages in either expanded or concise format. Servers configured with facility values of 0 - 3 receive syslog messages in concise format. Servers configured with facility values of 4 - 7 receive syslog messages in fully-expanded format. The default value is 0. The facility number used in syslog messages also identifies the origin of the message to the receiving server. You can use a syslog manager to view the syslog messages that the system sends. The system uses the User Datagram Protocol (UDP) to transmit the syslog message. You can specify up to a maximum of six syslog servers. You can use the management GUI or the command-line interface to configure and modify your syslog settings.

The syslog event notification settings apply to the entire system. You can specify the types of events that cause the system to send a notification. The possible types of notifications are error, warning, or information.

To specify a syslog destination, you *must* provide a valid IP address.

Note: Servers that are configured with facility values of 0 - 3 receive syslog messages in concise format. Servers that are configured with facility values of 4 - 7 receive syslog messages in fully expanded format.

To configure and work with notification settings, use the following commands:

Procedure

1. Issue the **mksyslogserver** CLI command to specify the action that you want to take when a syslog error or event is logged to the error log. For example, you can issue the following CLI command to set up a syslog notification:

   ```
   mksyslogserver syslog_server_name -ip 9.11.255.123
   ```

   where *syslog_server_name* is the name given to the Syslog server definition and *9.11.255.123* is the external Internet Protocol (IP) address of the syslog server.
2. To modify a syslog notification, issue the **chsyslogserver** command. For example:

   ```
   chsyslogserver syslog_server_name -ip 9.11.255.123
   ```

   where *syslog_server_name* is the name given to the Syslog server definition and *9.11.255.123* is the external IP address of the syslog server.
3. To delete a syslog notification, issue the **rmsyslogserver** command. For example:

   ```
   rmsyslogserver syslog_server_name -force
   ```
4. To display either a concise list or a detailed view of syslog servers that are configured on the system, issue the **lssyslogserver** command. For example, to display a concise view, enter the following command:

   ```
   lssyslogserver -delim :
   ```

   To display a detailed view of a syslog server, enter the following command:

   ```
   lssyslogserver syslog_server_name
   ```

Setting up email event notifications and inventory reports by using the CLI

You can use the command-line interface (CLI) to set up your system to send event notification and inventory reports to specified recipients and your support center.

Before you begin

You can configure Call Home by using the CLI.

About this task

To set up, manage, and activate email event, inventory, and Call Home notifications, complete the following steps:

Procedure

1. Enable your system to use the email notification function. To enable email notification, use the **mkemailserver** CLI command. Up to six SMTP email servers can be configured to provide redundant access to the external email network.

This example creates an email server object. It specifies the name, IP address, and port number of the SMTP email server. After you enter the command, you see a message that indicates that the email server was successfully created.

```
mkemailserver -ip ip_address -port port_number
```

where *ip_address* specifies the IP address of a remote email server and *port_number* specifies the port number for the email server.

2. Add recipients of email event and inventory notifications to the email event notification facility. To add recipients, use the **mkemailuser** CLI command.

   The following example adds email recipient *manager2008* and designates that this recipient is to receive email error-type event notifications.

   ```
   mkemailuser -address manager2008@ibm.com
    -error on -usertype local
   ```

   Important: Always select the *local* user type unless otherwise instructed by your support center. The *support* user type is normally only used with the Call Home feature.

   Remember: To control the frequency of email notifications, enter the following command: `chsystem -inventoryemail`

3. Set the contact information that is used by the email event notification facility. To set contact information, use the **chemail** CLI command. If you are starting the email event notification facility, the **-reply** parameter must be set.

   The following example sets the contact information for the email recipient *manager2008*.

   ```
   chemail -reply manager2008@ibm.com -contact manager2008
    -primary 0441234567 -location 'room 256 floor 1 IBM'
   ```

4. Optionally, generate a report that lists email event notification settings for all email recipients, or change or delete email recipients.
   - To generate a report that lists the email event notification settings for all email recipients, an individual email recipient, or a specified type of email recipient (local or support), use the **lsemailuser** CLI command.
   - To change the settings that are defined for a recipient, use the **chemailuser** CLI command. You must specify the user ID or name of the email recipient for whom you are modifying settings.
   - To remove a previously defined email recipient, use the **rmemailuser** CLI command. You must specify the user ID or name of the email recipient that you want to remove.

5. Activate the email and inventory notification function. To start the email and inventory notification function, use the **startemail** CLI command. The **startemail** command takes no parameters.

   Note: Inventory information is automatically reported to service personnel when you activate error reporting.

6. Optionally, test the email notification function to ensure that it is operating correctly and send an inventory email notification. The system uses the notifications settings to call home if errors occur.
   - To send a test email notification to one or more recipients, use the **testemail** CLI command. You must either specify **all** or the user ID or user name of an email recipient that you want to send a test email to.
   - To send an inventory email notification to all recipients that are enabled to receive inventory email notifications, use the **sendinventoryemail** CLI command. The **sendinventoryemail** command takes no parameters.
   - Use the **stopemail** command to stop the email and inventory notification function. The **stopemail** command takes no parameters.

Setting up email servers by using the CLI

You can set up email server objects by using the command-line interface (CLI).

About this task

You can specify a server object that describes a remote Simple Mail Transfer Protocol (SMTP) email server to receive event notifications from the clustered system. You can specify up to six servers to receive notifications. To configure and work with email servers, use the following commands:

Procedure

1. Use the **mkemailserver** CLI command to create an email server object that describes a remote Simple Mail Transfer Protocol (SMTP) email server. For example, enter the following CLI command to set up an email server:

   ```
   mkemailserver -ip ip_address
   ```

   where *ip_address* is the IP address of a remote email server. This address must be a valid IPv4 or IPv6 address.
2. To change the parameters of an existing email server object, use the **chemailserver** command. For example, to change the parameters of an email server, enter the following command:

   ```
   chemailserver -ip ip_address email_server_name_or_id
   ```

   where *ip_address* is the IP address of the email server object and *email_server_name_or_id* is the name or ID of the server object to be changed.
3. To delete a specified email server object, use the **rmemailserver** command. For example, to delete an email server, enter the following command:

   ```
   rmemailserver email_server_name_or_id
   ```
4. To display either a concise list or a detailed view of email servers that are configured on the system, use the **lsemailserver** command. For example, to display a concise view, enter the following command:

   ```
   lsemailserver -delim :
   ```

   To display a detailed view of an email server, enter the following command:

   ```
   lsemailserver email_server_name_or_id
   ```

Changing user passwords using the CLI

You can use the command-line interface (CLI) to change user passwords.

About this task

Passwords control access to these applications:

- System management GUI
- Service assistant GUI
- CLI

Follow these steps to change the password for a user:

Procedure

Enter the following command to change the password:

```
chuser -password cleartextpassword janedoe
```

Where *password* is the new password that you want to use for the user `janedoe`.

What to do next

Changing the locale setting using the CLI

You can use the command-line interface (CLI) to specify the locale for a system. The language that you select as your locale setting is used to display command results and error messages in the CLI.

About this task

The following locales are available:

- 0 US English (default)
- 3 Japanese

Procedure

Issue the **setlocale** CLI command with the ID for the locale.

Example

For example, issue the following CLI command to change the locale setting from US English to Japanese:

```
setlocale -locale 3
```

where *3* is the ID for the Japanese locale setting.

Viewing the feature log using the CLI

You can use the command-line interface (CLI) to view the feature log.

About this task

Perform the following steps to view the feature log:

Procedure

1. Issue the **lsdumps** command to return a list of dumps in the `/dumps/feature` destination directory. The feature log is maintained by the cluster. The feature log records events that are generated when license parameters are entered or when the current license settings have been breached.
2. Issue the **lsdumps** command to return a list of the files that exist of the type specified on the given node.

Analyzing the error log using the CLI

You can use the command-line interface (CLI) to analyze the error log (event log).

About this task

Perform the following step to analyze the error log:

Procedure

Issue the following CLI command to list error log entries by file type: **lseventlog**

Results

This command lists the error log entries. You can filter by type; for example, `lseventlog -filtervalue object_type=mdisk` displays the error log by managed disks (MDisks).

You can display the whole log or filter the log so that only errors, events, or unfixed errors are displayed. You can also request that the output is sorted either by error priority or by time. For error priority, the most serious errors are the lowest-numbered errors. Therefore, the most serious errors are displayed first in the table. For time, either the older or the latest entry can be displayed first in the output.

Shutting down a system by using the CLI

You can use the command-line interface (CLI) to shut down a system.

Procedure

To power off your system, complete the following steps.

1. Determine which hosts have access to volumes on this system by running the **`lshostvdiskmap`** command.
2. Stop input/output (I/O) to the system from each host that is listed in step 1.

   Note: Failure to stop host I/O can result in failed I/O operations being reported to your host operating systems.
3. Shut down the system by using this command:
   ```
   stopsystem
   ```
4. Wait for the power light-emitting diodes (LEDs) on all nodes to flash at 1 Hz, indicating that the shutdown operation has completed.

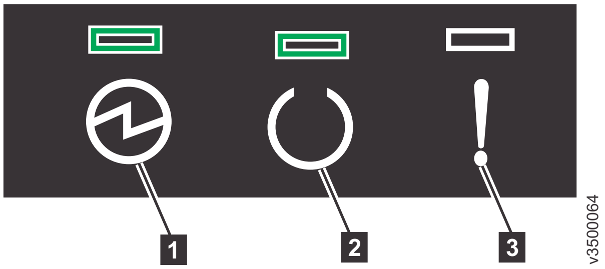


Figure 1. Location of the power LED on a node canister

   1 Power
   2 Status
   3 Fault
5. Disconnect the power cords from both power supplies in each node.
6. Disconnect the power cords from both power supplies in each expansion enclosure.

Updating the system automatically using the CLI

You can use the command-line interface (CLI) to install software updates.

Before you begin

Follow these steps to update to version 8.1.0 or later from version 7.7.0 or later.

To update from version 5.1.x or earlier, see the relevant IBM Knowledge Center or publications that are available at this website: www.ibm.com/support

If you encounter a memory DIMM failure to any node during the update process, stop immediately and follow directions in Updating the system.

You can use the command-line interface to resolve multipathing issues when nodes go offline for updates. You can add the ability to override the default 30 minute mid-point delay, pause an update, and resume a stalled update by following these steps:

1. To start an update but pause at the halfway point, enter the following command:
   ```
   applysoftware -file filename -pause
   ```
2. To start an update but then pause before you take the node offline for an update, enter the following command:
   ```
   applysoftware -file filename -pause -all
   ```
3. To resume a stalled update and pause at the halfway point, enter the following command:
   ```
   applysoftware -resume -pause
   ```
4. To resume a stalled update and pause before you take the remaining nodes offline for an update, enter the following command:
   ```
   applysoftware -resume -pause -all
   ```

   Note: The **-all** parameter enables the update to pause indefinitely before each node goes offline for an update. This pause happens before the existing object-dependent volume check is carried out. The **-resume** parameter enables the user to continue the update.

About this task

To update the system, follow these steps.

Procedure

1. You must download, install, and run the latest version of the test utility to verify that no issues exist with the current system.

   Important: After you install and run the test utility by using either the management GUI or the CLI, you must return to step 2 on this page.

   You can download and install the most current version of this tool at the following website. The link provides instructions for using either the management GUI or the CLI to install and run the test utility.

   http://www.ibm.com/support/docview.wss?uid=ssg1S4000585
2. Download the latest code from the www.ibm.com/support site.
   - If you want to write the code to a CD, you must download the CD image.
   - If you do not want to write the code to a CD, you must download the installation image.
3. Use PuTTY scp (pscp) to copy the update files to the node.
4. Ensure that the update file was successfully copied.

   Before you begin the update, you must be aware of the following situations:
   - The installation process fails under the following conditions:
     - If the code that is installed on the remote system is not compatible with the new code or if an intersystem communication error does not allow the system to check that the code is compatible.
     - If any node in the system has a hardware type that is not supported by the new code.
     - If the system determines that one or more volumes in the system would be taken offline by rebooting the nodes as part of the update process. You can find details about which volumes

would be affected by using the **lsdependentvdisks** command. If you are prepared to lose access to data during the update, you can use the force flag to override this restriction.
- The update is distributed to all the nodes in the system by using internal connections between the nodes.
- Nodes are updated one at a time.
- Nodes run the new code concurrently with normal system activity.
- While the node is updated, it does not participate in I/O activity in the I/O group. As a result, all I/O activity for the volumes in the I/O group is directed to the other node in the I/O group by the host multipathing software.
- There is a thirty-minute delay between node updates. The delay allows time for the host multipathing software to rediscover paths to the nodes that are updated. There is no loss of access when another node in the I/O group is updated.
- The update is not committed until all nodes in the system are successfully updated to the new code level. If all nodes are successfully restarted with the new code level, the new level is committed. When the new level is committed, the system vital product data (VPD) is updated to reflect the new code level.
- Wait until all member nodes are updated and the update is committed before you invoke the new functions of the updated code.
- Because the update process takes some time, the installation command completes as soon as the code level is verified by the system. To determine when the update is completed, you must either display the code level in the system VPD or look for the **Software update complete** event in the error/event log. If any node fails to restart with the new code level or fails at any other time during the process, the code level is backed off.
- During an update, the version number of each node is updated when the code is installed and the node is restarted. The system code version number is updated when the new code level is committed.
- When the update starts, an entry is made in the error or event log and another entry is made when the update completes or fails.

5. Issue this CLI command to start the update process:

   ```
   applysoftware -file software_update_file
   ```

   Where *software_update_file* is the name of the code update file in the directory you copied the file to in step 3 on page 85.If the system identifies any volumes that would go offline as a result of rebooting the nodes as part of the system update, the code update does not start. An optional **force** parameter can be used to indicate that the update continues regardless of the problem identified. If you use the **force** parameter, you are prompted to confirm that you want to continue. The behavior of the **force** parameter changes, and it is no longer required when you apply an update to a system with errors in the event log.
6. If you are updating from a release before version 7.4.0, issue the following CLI command to check the status of the code update process:

   ```
   svcinfo lssoftwareupgradestatus
   ```

   This command displays `inactive` when the update is complete.

   Note: If a status of `stalled_non_redundant` is displayed, proceeding with the remaining set of node updates might result in offline volumes. Contact a service representative to complete the update.
7. If you are updating from version 7.4.0 or later, issue the following CLI command to check the status of the code update process:

   ```
   lsupdate
   ```

   This command displays `success` when the update is complete. If you have hot-spare nodes that are configured on your system, the hot-spare node assumes I/O operations from each node as it is updated.

Note: If a status of `stalled_non_redundant` is displayed, proceeding with the remaining set of node updates might result in offline volumes. Contact a service representative to complete the update.

8. If you updated from a release before version 7.4.0, you receive the status message `system_completion_required`. To complete the update process, issue the command **`applysoftware -complete`**. After that command is run, you can run **`lsupdate`** to see the progress of the update completion.
9. To verify that the update successfully completed, issue the **`lsnodevpd`** CLI command for each node that is in the system.

   The code version field displays the new code level.

   Important: If you update your system software to version 8.1.1 or later from a version earlier than 8.1.0, on a system where you have already installed more than 64 GB of RAM, all nodes return from the update with an error code of 841. Versions 8.1.0 and later allocate memory in a different way than previous versions, so the RAM must be "accepted" again. To resolve the error, complete the following steps:

   a. On a single node, run the **`svctask chnodehw`** command. Do not run the command on more than one node at a time.
   b. Wait for the node to restart and return without the error.
   c. Wait an additional 30 minutes for multipath drives to recover on the host.
   d. Repeat this process for each node individually until you clear the error on all nodes.

Results

When a new code level is applied, it is automatically installed on all the nodes that are in the system.

Note: An automatic system update can take up to 30 minutes per node.

Chapter 4. Array commands

Use the array commands to manage arrays and their properties.

charray

Use the **charray** command to change array attributes.

Syntax

```
►►─charray─┬──────────────────────────┬─ ─┬──────────────────────┬─ ─┬─────────────┬─ ──────────────►
           └─ -name ── new_name_arg ──┘   └─ -sparegoal ──1-100──┘   └─ -balanced ─┘

►─┬─────────────────────────────────────────────┬─ ─ ─┬──────────────────────────┬─┬─mdisk_id───┬──────────►◄
  └─ -slowwritepriority ──┬─latency────┬────────┘     └─-rebuildareasgoal──┬─0─┬─┘ └─mdisk_name─┘
                          └─redundancy─┘                                   ├─1─┤
                                                                           ├─2─┤
                                                                           ├─3─┤
                                                                           └─4─┘
```

Parameters

-name *new_name_arg*
(Optional) Specifies the new name to apply to the array MDisk.

-sparegoal 1-100
(Optional) Sets the number of spares to protect the array members with. The value can be a number between 1 and 100.

Note: This parameter is not applicable for distributed arrays.

-balanced
(Optional) Forces the array to balance and configure the spare goals of the present drives.

Specify **-balanced** and the system examines the membership's chain balance for mirrored arrays. If each mirrored member is on a different chain than its partner member, the array continues balancing the member chains. If each mirrored member is not on a different chain than its partner member, the array stops balancing the member chains.

Note:

- If **-balanced** is specified and the goal of the associated array MDisk changes, the tier of the array MDisk is updated to match the new goal.
- This parameter is not applicable for distributed arrays.

-slowwritepriority latency | redundancy
(Optional) Controls array ability to complete write operations that take too long, even if it temporarily compromises redundancy.

The value can be either latency or redundancy:

- latency implies the feature is enabled for normal I/O operations
- redundancy implies the feature is not enabled for normal I/O operations

The default value is latency mode for existing arrays, unless the array is RAID-0 (in which case redundancy mode is required).

Important: Do not change the mode of a RAID-0 array.

Important: An array can cause member drives to become unsynchronized (to preserve response time) if the value is `latency`. If the value is `redundancy`, the array cannot cause member drives to become unsynchronized (to preserve time) and I/O performance is impacted.

-rebuildareasgoal *0* | *1* | *2* | *3* | *4*

(Optional) Specifies the rebuild areas threshold. The array logs an error when the available rebuild areas drop below this specified threshold. The values are `0`, `1`, `2`, `3`, or `4`. (If you specify `0`, an error is not logged if the system runs out of rebuild areas.)

Note: This parameter is only applicable for distributed arrays.

mdisk_id | *mdisk_name*

(Required) Identifies (by ID or user-defined name) which array the MDisk command applies to.

Description

This command changes an array's attributes.

An invocation example to change the name of an array

```
charray -name raid6 mdisk0 0
```

The resulting output:

```
No feedback
```

An invocation example to set the number of spares threshold to 2

```
charray -sparegoal 2 mdisk52
```

The resulting output:

```
No feedback
```

An invocation example to balance the array

```
charray -balanced 3
```

The resulting output:

```
No feedback
```

An invocation example for changing the rebuild areas goal for an array

```
charray -rebuildareasgoal 3 array1
```

The resulting output:

```
No feedback
```

An invocation example for changing the rebuild areas goal for an array

```
charray -slowwritepriority redundancy 0
```

The resulting output:

```
No feedback
```

charraymember

Use the **charraymember** command to modify an array member's attributes, or to swap (exchange) a member of a RAID array with that of another drive.

Syntax

```
►►──charraymember── -member ──member_id── ──┬─ -balanced ──────────────┬── ──┬─ mdisk_id ───┬──────────►◄
                                            ├─ -newdrive ──new_drive_id─┤     └─ mdisk_name ─┘
                                            ├─ -immediate ──────────────┤
                                            └─ -unbalanced ─────────────┘
```

Parameters

-member *member_id*
Identifies the array member index.

-balanced
(Optional) Forces the array member spare goals to be set to the:

- Present array member goals
- Existing exchange goals
- The newDrive goals

Note: If **-balanced** is specified and the goal of the associated array MDisk changes, the tier of the array MDisk is updated to match the new goal.

-newdrive *new_drive_id*
(Optional) Identifies the drive to add to the array.

For distributed arrays:

- If the **-newdrive** parameter is specified with the **-immediate** parameter, and the member is not associated with a rebuild area, the command begins a distributed rebuild to a rebuild area in such a way that a copyback begins immediately when the rebuild finishes.
- If the **-newdrive** parameter is specified and the member is already associated with a rebuild area, the array configures itself to use the new member, and might begin a copyback. (This occurs whether or not the **-immediate** parameter is specified.)
- If the **-immediate** parameter is not set and the **-newdrive** parameter is set (but the array member is not allocated to a rebuild area), the command fails. If the **-balanced** parameter is set, the command fails.

-immediate
(Optional) Specifies that the old disk is to be immediately removed from the array, and the new disk rebuilt. If you do not choose this option, exchange is used; this preserves redundancy during the rebuild.

-unbalanced
(Optional) Forces the array member to change if the newDrive does not meet array member goals.

mdisk_id
(Either the ID or the name is required) Identifies which ID array the MDisk command applies to.

mdisk_name
(Either the ID or the name is required) Identifies which name array the MDisk command applies to.

Description

This command modifies an array member's attributes, or to swap a member of a RAID array with that of another drive.

Specify **-balanced** and the system examines the mirrored pair containing the member (including the new member drive's properties). If the array is mirror-based and the new drive is:

- On the same chain as the other member of this pair, it removes the chain-balancing goal from the array

- Not on the same chain as the other member of this pair (and there is only one mirrored pair) the array becomes chain-balanced

Because **charraymember** is member-focussed this command only operates locally to the member being operated on in terms of interacting with the new chain balanced goal.

Table 10 shows the command combination options.

Table 10. charraymember combination options

Option	Description
-balanced	• Member goals are set to the properties of the existing member or exchange drive. • The command will fail if the member is not populated with a drive. • Member goals are set to the properties of the current member drives being exchanged into the array count as members. • If no exchange exists, the existing member drive goals are used.
-newdrive drive_id	• The command processes the exchange, and does NOT update the member goals. • You must specify a new drive that is an exact match for the member goals. • The command will fail if the drive is not an exact match.
-newdrive drive_id -balanced	The command processes the exchange and updates the member goals to the properties of the new drive.
-newdrive drive_id -unbalanced	• The command processes the exchange and does NOT update the member goals. • This is only permitted when the array is degraded and the member is empty. • This means **-immediate** is mute, the exchange is always immediate. • Later, if drives are a sufficient member goal match, the array rebalance selects those drives. • A balancing exchange restarts the member goals.

An invocation example to swap a spare or candidate drive for a member 0 drive by using exchange

```
charraymember -member 0 -newdrive 4 mdisk2
```

The resulting output:

```
No feedback
```

An invocation example to swap a spare or candidate drive for a member 1 drive and start component rebuild for the new member

```
charraymember -member 1 -newdrive 3 -immediate mdisk3
```

The resulting output:

```
No feedback
```

An invocation example to swap in a spare or candidate drive for member index 2

If there is a drive present the exchange occurs:

```
charraymember -member 2 -newdrive 4 mdisk4
```

The resulting output:

```
No feedback
```

An invocation example to force member 4 to change its spare goals to its associated drive

```
charraymember -member 4 -balanced mdisk6
```

The resulting output:

```
No feedback
```

An invocation example to force an exchange and make the array change its goals to the new drive

```
charraymember -member 3 -newdrive 9 -balanced mdisk5
```

The resulting output:

```
No feedback
```

An invocation example to force an unbalancing exchange when drive 8 does not match the goals

```
charraymember -member 2 -newdrive 8 -unbalanced mdisk5
```

The resulting output:

```
No feedback
```

An invocation example to force an immediate exchange and make the array change its goals to the new drive

```
charraymember -member 3 -newdrive 9 -balanced -immediate mdisk5
```

The resulting output:

```
No feedback
```

An invocation example to change member 24 for new drive 15 by using a distributed rebuild to a rebuild area

```
charraymember -member 24 -newdrive 15 -immediate 0
```

The resulting output:

```
No feedback
```

lsarray

Use the **lsarray** command to list the array MDisks.

Syntax

```
►►─ lsarray ─ ─┬──────────┬─ ─┬──────────────────────┬─ ─┬────────┬─ ─┬──────────────────┬──────►
               └─ -nohdr ─┘   └─ -delim ─ delimiter ─┘   └-bytes──┘   └─ -filtervalue? ─┘

►─ ─┬────────────────────────────────────────────────────┬─ ─┬───────────────┬─ ─────────────►◄
    ├─ -filtervalue ─ attribute=value ───────────────────┤   └─┬ mdisk_id ───┬┘
    └─ -filtervalue ─ capacity=value ─ -unit ──┬─ b ──┬──┘     └ mdisk_name ─┘
                                               ├─ kb ─┤
                                               ├─ mb ─┤
                                               ├─ gb ─┤
                                               ├─ tb ─┤
                                               └─ pb ─┘
```

Parameters

-nohdr

(Optional) By default, headings are displayed for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*

(Optional) In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. In a detailed view, the data is separated from its header by the specified delimiter.

-bytes

(Optional) Requests output of capacities in bytes (instead of rounded values).

-filtervalue?

(Optional) Includes all of the valid filter attributes in the report. The following filter attributes are valid for the **lsarray** command:

- balanced
- capacity
- distributed
- mdisk_id
- mdisk_name
- mode
- mdisk_grp_id
- mdisk_grp_name
- fast_write_state
- raid_status
- raid_level
- redundancy
- spare_goal
- spare_protection_min
- status
- strip_size
- tier
- easy_tier_load

Any parameters that are specified with the **-filtervalue?** parameter are ignored.

-filtervalue *attribute=value*

(Optional) Specifies a list of one or more filter attributes match the specified values; see **-filtervalue?** for the supported attributes. Only objects with a value that matches the filter attribute value are returned. If `capacity` is specified, the units must also be included. Use the **unit** parameter to interpret the value for size or capacity.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards when you use the CLI:

- The wildcard character is an asterisk (*).
- The command can contain a maximum of one wildcard, which must be the first or last character in the string.
- When you use a wildcard character, you must enclose the filter entry within double quotation marks (""):

```
lsarray -filtervalue  "name=md*"
```

-filtervalue *capacity=value*
(Optional) Specifies a list of one or more filter capacities (size) values matching the specified values for the **unit** parameter. Use the **unit** parameter to interpret the value for size or capacity.

-unit *b* | *kb* | *mb* | *gb* | *tb* | *pb*
(Optional) The units that are used when you specify the **-filtervalue** capacity, where:

- *b* = bytes
- *kb* = 1,024 bytes
- *mb* = 1,048,576 bytes
- *gb* = 1,073,741,824 bytes
- *tb* = 1,099,511,627,776 bytes
- *pb* = 1,125,899,906,842,624 bytes

mdisk_id
(Optional) The identity of the array MDisk.

mdisk_name
(Optional) The name of the array MDisk.

Description

This command returns a concise list or a detailed view of array MDisks visible to the clustered system (system). This table provides the attribute values that can be displayed as output view data.

Table 11. Array output

Attribute	Values
capacity	Indicates the value for the capacity you specify by using the **-unit** parameter.
status	• online • offline • excluded • degraded (applies only to internal MDisks)
mode	Indicates the mode. The values are: • `unmanaged` • `managed` • `image` • `array`
quorum_index	Indicates the quorum index. The values are: • `0` • `1` • `2` • Blank if the MDisk is not being used as a quorum disk
block_size	Indicates the block size. The value is `512` bytes (or blank) in each block of storage.
ctrl_type	`4`, `6`, where `6` is a flash drive attached inside a node and `4` is any other device

Table 11. Array output (continued)

Attribute	Values
raid_status	Indicates the RAID status. The values are: **offline** The array is offline on all nodes. **degraded** The array has deconfigured or offline members; the array is not fully redundant. **syncing** The array members are all online. The array is synchronizing parity or mirrors to achieve redundancy. **initializing** The array members are all online. The array is initializing; the array is fully redundant. **online** The array members are all online, and the array is fully redundant.
fast_write_state	Indicates the cache state of the array. The values are: • `empty`, which indicates that the array disk data is not changing • `not_empty`, which indicates that the array disk data might change • `corrupt`, which indicates that the array disk data is lost and the array is corrupt Repair can be initiated by using the **`recoverarray`** or **`recoverarraybysystem`** command.
raid_level	Indicates the RAID level of the array. The values are: • `RAID0` • `RAID1` • `RAID5` • `RAID6` • `RAID10`
redundancy	Indicates the number of member disks that can fail concurrently without causing the array to fail.
strip_size	Indicates the strip size of the array (in KB).
spare_goal	Indicates the number of spares that the array members must be protected by. For distributed arrays, this value is blank.
spare_protection_min	Indicates the minimum number of spares that an array member is protected by. For distributed arrays, this value is blank.
balanced	For nondistributed arrays, this value indicates whether the array is balanced to its spare goals: • `exact` indicates that all populated members have the same (matching) capability and location. • `yes` indicates that all populated members have at least the same capability and chain, but a different enclosure or slot. • `no` indicates that it is unbalanced. For distributed arrays, this value indicates whether a superior drive class is being used for the array: • `exact` indicates that the same drive class is in use. • `yes` indicates that at least one array member exceeds the array drive class. For distributed arrays, the array must also be balanced to its rebuild area goals.

Table 11. Array output (continued)

Attribute	Values
tier	Indicates the tier that this array is assigned to by auto-detection (for internal arrays) or by the user: • tier0_flash • tier1_flash • tier_enterprise • tier_nearline **Note:** Use the **chmdisk** command to change this value.
replacement_date	Indicates the date of a potential array failure. The format must be YYMMDD.
easy_tier_load	Indicates the value for Easy Tier settings, and is either blank (for arrays) or one of the following values (for MDisks): • low • medium • high • very_high
slow_write_priority	Indicates the response time goal: • latency indicates that the array is taken out of synchronization to quickly complete write operations that take excessive time. • redundancy indicates slow write operations are completed in normal time and the arrays remain synchronized.
site_id	Indicates the site value for the storage pool. This numeric value is 1, 2, 3, or blank.
site_name	Indicates the site name for the storage pool. This value is alphanumeric or is blank.
fabric_type	Indicates a Fibre Channel (FC), SAS, or another type of array. • fc indicates an array from an FC controller • sas_direct indicates an array from an SAS direct-attached controller
encrypt	Indicates whether the data that is stored on the array is encrypted or not encrypted. The possible values are: • yes • no
distributed	Indicates whether the array is distributed. The values are yes or no.
drive_class_id	Indicates the drive class that makes up this array. If **-allowsuperior** was used during array creation, the lowest used drive class ID is displayed. This value is blank for nondistributed arrays.
drive_count	Indicates the total width of the array, including rebuild areas. The value is a number from 4 to 128. The minimum value for RAID-6 and RAID-10 arrays is 6.
stripe_width	Indicates the width of a single unit of redundancy within a distributed set of drives. The values are: • Any number from 3 - 16 for RAID-5 arrays • Any number from 4 - 16 for RAID-6 arrays • An even number from 2 - 16 for RAID-10 arrays
rebuild_areas_total	Indicates the total number of rebuild areas set when the array is created. These rebuild areas provide performance but no capacity. The value is 1 - 4 for distributed array RAID-5 and distributed array RAID-6. The value is blank for nondistributed arrays.
rebuild_areas_available	Indicates the number of remaining rebuild areas within the set of drives. The value is 1 - 4 for distributed array RAID-5 and distributed array RAID-6. The value is blank for nondistributed arrays.

Table 11. Array output (continued)

Attribute	Values
rebuild_areas_goal	Indicates the rebuild areas threshold (minimum limit) at which point the array logs an error. The value is 1 - 4 for distributed array RAID-5 and distributed array RAID-6. The value is blank for nondistributed arrays.

This list defines the status fields:

online The MDisk is online and available.

degraded
(Internal MDisks only) The array has members that are degraded, or the raid_status is degraded.

degraded_ports
There are one or more MDisk port errors.

degraded_paths
One or more paths to the MDisk are lost; the MDisk is not online to every node in the system.

offline
All paths to the MDisk are lost.

excluded
The MDisk is excluded from use by the system; the MDisk port error count exceeded the threshold.

A concise invocation example

```
lsarray -delim :
```

The resulting output:

```
mdisk_id:mdisk_name:status:mdisk_grp_id:mdisk_grp_name:capacity:raid_status:
raid_level:redundancy:strip_size:tier:encrypt
:distributed
1::online:0:mdiskgrp0:68.4GB:online:raid0:0:256:enterprise:no:yes
2:mdisk2:online:0:mdiskgrp0:88.4GB:syncing:raid5:1:256:nearline:no:no
533:mdisk533:degraded:1:mdiskgrp1:78.2GB:syncing:raid6:2:128:ssd:yes:yes
534:mdisk534:online:2:mdiskgrp1:94.2GB:initting:raid6:2:64:ssd:yes:no
```

A detailed invocation example

```
lsarray
```

The resulting output:

```
mdisk_id 144
mdisk_name draid6_5
status online
mode array
mdisk_grp_id 1
mdisk_grp_name pool_512
capacity 5.6TB
quorum_index
block_size
controller_name
ctrl_type
ctrl_WWNN
controller_id
path_count
max_path_count
ctrl_LUN_#
UID
preferred_WWPN
```

```
active_WWPN
fast_write_state not_empty
raid_status initting
raid_level raid6
redundancy 2
strip_size 256
spare_goal
spare_protection_min
balanced exact

tier tier0_flash
replacement_date 121110090907
slow_write_priority redundancy
fabric_type
site_id
site_name
easy_tier_load
encrypt no
distributed yes
drive_class_id 1
drive_count 28
stripe_width 15
rebuild_areas_total 4
rebuild_areas_available 4
rebuild_areas_goal 2
```

A detailed invocation example

```
lsarray 1
```

The resulting output:

```
mdisk_id:1
mdisk_name:
status:online
mode:array
mdisk_grp_id:0
mdisk_grp_name:mdiskgrp0
capacity:68.4GB
quorum_index:
block_size:
controller_name:
ctrl_type:
ctrl_WWNN:
controller_id:
path_count:
max_path_count:
ctrl_LUN_#:
UID:
preferred_WWPN:
active_WWPN:
fast_write_state:empty
raid_status:online
raid_level:raid0
redundancy:0
strip_size:256
spare_goal:2
spare_protection_min:2
balanced:yes


tier tier1_flash
replacement_date 121110090907
slow_write_priority:latency
site_id:3
site_name:Quorum
fabric_type:
encrypt:yes
```

```
distributed no
 drive_class_id
 drive_count 8
 stripe_width 4
 total_rebuild_areas
 available_rebuild_areas
 rebuild_areas_goal
```

lsarrayinitprogress

Use the **lsarrayinitprogress** command to view the progress of array background initialization that occurs after creation.

Syntax

```
►►── lsarrayinitprogress ──┬──────────┬── ┬──────────────────────────────────┬── ──────────────►
                           └─ -nohdr ─┘   └─ -filtervalue ── attribute_value ─┘

►──┬──────────────────┬── ┬─────────────────────┬── ┬─────────────┬──────────────────────────►◄
   └─ -filtervalue? ─┘    └─ -delim ──delimiter─┘   ├─mdisk id────┤
                                                    └─mdisk_name──┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards:

- The wildcard character is the asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, enclose the filter entry within double quotation marks (""):
  `lsarraysyncprogress -filtervalue mdisk_id="1*"`

-filtervalue?
(Optional) Displays the valid filter attributes for the **-filtervalue** parameter:

- estimated_completion_time
- mdisk_id
- mdisk_name
- progress

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

mdisk_id
(Optional) The identity of the array MDisk.

mdisk_name
(Optional) The user-defined MDisk name.

Description

This command shows the progress of array background initialization. Table 12 shows possible outputs.

Table 12. lsarrayinitprogress output

Attribute	Value
progress	The percentage of initialization task that is completed.
estimated_completion_time	The expected initialization task completion time, in YYMMDDHHMMSS format.

A concise invocation example

```
lsarrayinitprogress –delim :
```

The resulting output:

```
mdisk_id:mdisk_name:progress:estimated_completion_time
0:mdisk0:50:070301120000
1:mdisk1:51:070301130000
2:mdisk2:32:070301153500
```

A concise invocation (qualified with MDisk) example

```
lsarrayinitprogress –delim : mdisk2
```

The resulting output:

```
mdisk_id:mdisk_name:progress:estimated_completion_time
2:mdisk2:32:070301153500
```

An invocation example for an array that has finished initialization

```
lsarrayinitprogress –delim : mdisk4
```

The resulting output:

```
mdisk_id:mdisk_name:progress:estimated_completion_time
4:mdisk4:100:
```

lsarraylba

Use the **lsarraylba** command to permit an array logical block address (LBA) to be found from a drive and LBA.

Syntax

```
►►── lsarraylba ──┬─────────┬── ┬────────────────────┬── ──────────────────────────►
                  └─ -nohdr ┘   └─ -delim ─ delimiter ┘

►─ -drivelba ── lba ── -drive ── drive_id ── ─────────────────────────────────────►◄
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-drivelba *lba*
The LBA on the drive to convert to the array LBA. The *lba* value must be specified in hex, with a `0x` prefix.

-drive *drive_id*
The ID of the drive to view.

Description

This command permits an array LBA to be found on a drive and LBA.

The system provides volumes that have LBAs for 512-byte block sizes; however, back-end disks that have a block size of either 512 or 4096 bytes can also be used. Drives are listed in their physical size.

Use the **lsdrive** command to display the drive block size, and use the **lsdrive** or `lsarray` command to list each object (the drive and the MDisk).

Table 13 shows possible outputs.

Table 13. lsarraylba output

Attribute	**Value**
type	The type of MDisk extent allocation: • `allocated` • `unallocated` For distributed arrays only: • If the LBA is an unused rebuild area, this value displays `rebuild_area`. • If the LBA is a used rebuild area, this value displays `allocated`.
mdisk_lba	The LBA on the array MDisk (blank if none).
mdisk_start	The start of range of LBAs (strip) on the array MDisk (blank if none).
mdisk_end	The end of range of LBAs (strip) on the array MDisk (blank if none).
drive_start	The start of range of LBAs (strip) on the drive (blank if none).
drive_end	The end of range of LBAs (strip) on the drive (blank if none).

An invocation example to map drive `2` LBA `-xff` to MDisk `2` LBA `0xff`

```
lsarraylba -delim : -drivelba 0xff -drive 2
```

The resulting output:

```
mdisk_id:mdisk_name:type:mdisk_lba:mdisk_start:mdisk_end:drive_start:drive_end
0:mdisk2:allocated:0x00000000000001ff:0x0000000000000100:0x00000000000001ff:
0x0000000000000000:0x00000000000000ff
```

An invocation example for an allocated space

```
lsarraylba -drivelba 0x00 -drive 2
```

The resulting output:

```
mdisk_id mdisk_name type      mdisk_lba          mdisk_start        mdisk_end          drive_start        drive_end
1        mdisk1     allocated 0x0000000000000000 0x0000000000000000 0x00000000000001FF 0x0000000000000000 0x000000000000
```

An invocation example for an unused rebuild area

```
lsarraylba -drivelba 0x00 -drive 16
```

The resulting output:

```
mdisk_id mdisk_name type         mdisk_lba          mdisk_start        mdisk_end          drive_start        drive_end
3        mdisk3     rebuild_area                                                          0x0000000000000000 0x0000000000
```

lsarraymember

Use the **lsarraymember** command to list the member drives of one or more array MDisks.

Syntax

```
►►── lsarraymember ── ──┬──────────┬── ──┬──────────────────────────────────┬── ──────────────────────►
                        └─ -nohdr ─┘     └─ -filtervalue ── attribute=value ─┘

►──┬─────────────────┬── ──┬───────────────────────┬── ──┬──────────────┬──────────────────────────►◄
   └─ -filtervalue? ─┘     └─ -delim ── delimiter ─┘     └─┬─mdisk_id───┬┘
                                                           └─mdisk_name─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filter attributes that matches the specified values; see **-filtervalue?** for the supported attributes.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards when you use the CLI:

- The wildcard character is an asterisk (*).
- The command can contain a maximum of one wildcard, which must be the first or last character in the string.
- When you use a wildcard character, you must enclose the filter entry within double quotation marks (""):

```
lsarraymember -filtervalue  "mdisk_name=md*"
```

-filtervalue?
(Optional) Includes all of the valid filter attributes in the report. The following filter attributes are valid for the **lsarraymember** command:

- mdisk_id
- mdisk_name
- member_id
- drive_id
- new_drive_id
- spare_protection
- balanced

Any parameters specified with the **-filtervalue?** parameter are ignored.

-delim *delimiter*
(Optional) By default, in a concise view all columns of data are space-separated, with the width of each column set to the maximum width of each item of data. In a detailed view, each item of data is an individual row, and if headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. Enter `-delim :` on the command line, and the colon character (`:`) separates all items of data in a concise view (for example, the spacing of columns does not occur); in a detailed view, the specified *delimiter* separates the data from its header.

mdisk_id
(Optional) The identity of the array MDisk.

mdisk_name
(Optional) The MDisk name that you provided.

Description

This command lists the member drives of one or more array MDisks. It describes positions within an array unoccupied by a drive. The positions determine how mirroring the RAIDs takes place. For example, determining whether `x` is mirrored to `y` for RAID-10, where parity starts from RAID-5, which is for enclosure-based systems only.

Table 14 shows the potential output for this command.

Table 14. lsarraymember output

Attribute	Value
member_id	Specifies the identity of the array member. It represents drive order in RAID array
drive_id	Specifies the identity of the drive for member ID, or the source drive if an exchange is in progress. It is blank if there is no drive that is configured.
new_drive_id	Specifies the ID of the drive that is exchanged with this member ID. It is blank if there is no ID.
spare_protection	Specifies the number of non-degrading spares for the member. This includes spare drives with different attributes from the array member goals that perform equally or better than the array member goals. For distributed array members this field is blank.

Table 14. lsarraymember output (continued)

Attribute	Value
balanced	For nondistributed arrays, this value describes if the array is balanced to its spare goals: • exact indicates that all populated members have the same (matching) capability and location. • yes indicates that all populated members have at least the same capability and chain, but a different enclosure or slot. • no indicates that it is unbalanced. For distributed arrays, this value indicates whether a superior drive class is being used for the array: • exact indicates that the same drive class is in use. • yes indicates that the drive exceeds the array drive class. For distributed arrays, the array must also be balanced to its rebuild area goals.
slow_write_count	Indicates the number of times this member becomes unsynchronized because of high response time on write I/O operations.
slow_write_time_last	Creates a timestamp of when the component last became unsynchronized. The time format is `YYMMDDhhmmss` in clustered system time. No time is indicated if the value for `slow_write_count` is `0`.

A concise invocation example

```
lsarraymember -delim :
```

The resulting output:

```
lsarraymember -delim :
mdisk_id:mdisk_name:member_id:drive_id:new_drive_id:spare_protection
        :balanced:slow_write_count:slow_write_time_last
2:mdisk1:0:55::1:exact:4:130103202158
2:mdisk1:1:56::1:exact:1:130103203930
2:mdisk2:0:0::2:exact:0:
2:mdisk2:1:2:5:3:exact:2:130103204044
2:mdisk2:2:::::
2:mdisk2:3:8::0:no::
```

A concise invocation example (qualified with MDisk)

```
lsarraymember -delim : mdisk_2
```

The resulting output:

```
tmdisk_id:mdisk_name:member_id:drive_id:new_drive_id:spare_protection
         :balanced:slow_write_count:slow_write_time_last
2:mdisk2:0:0::2:exact:4:130103202158
2:mdisk2:1:2:5:3:exact:1:130103203930
2:mdisk2:2:::::0:
2:mdisk2:3:8::0:no:2:130103204044
```

Note: From this output, you can see that:

- The array has four members (possibly a 4-member RAID-10 array). You cannot use RAID-10 with distributed arrays.
- The second array member is undergoing exchange for `drive5`.
- The third array member is not configured. It might be offline or failed, without a hot spare available.
- The fourth array member has no spare protection and is not balanced.

An invocation example (two arrays)

```
lsarraymember -delim :
```

The resulting output:

```
mdisk_id:mdisk_name:member_id:drive_id:new_drive_id:spare_protection
        :balanced:slow_write_count:slow_write_time_last
2:mdisk1:0:55:::1:exact:4:130103202158
2:mdisk1:1:56:::1:exact:1:130103203930
2:mdisk2:0:0:::2:exact:0:
2:mdisk2:1:2:5::3:exact:2:130103204044
2:mdisk2:2:::::::
2:mdisk2:3:8:::0:no::
```

An invocation example (an array with a change in membership from (55,56) to (55,57,58))

```
lsarraymember -delim : mdisk_3
```

The resulting output:

```
mdisk_id:mdisk_name:member_id:drive_id:new_drive_id:spare_protection
        :balanced:slow_write_count:slow_write_time_last
3:mdisk3:0:55::55:1:exact:4:130103202158
3:mdisk3:1:56::57:1:exact:1:130103203930
3:mdisk3:2:::58:1:exact:0:
```

An invocation example (an array with a change in membership from (55,57,58) to (55,56))

```
lsarraymember -delim : mdisk_3
```

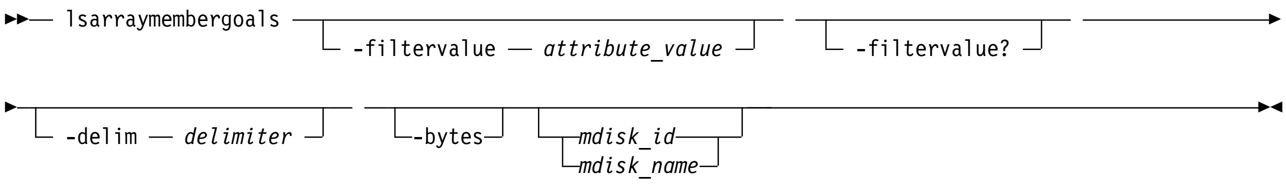
The resulting output:

```
mdisk_id:mdisk_name:member_id:drive_id:new_drive_id:spare_protection
        :balanced:slow_write_count:slow_write_time_last
3:mdisk3:0:55::55:1:exact:4:130103202158
3:mdisk3:1:57::56:1:exact:1:130103203930
3:mdisk3:2:58:::1:exact:0:
```

lsarraymembergoals

Use the **lsarraymembergoals** command to list the spare goals for member drives of one or more array MDisks.

Syntax



Parameters

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards:

- The wildcard character is the asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, enclose the filter entry within double quotation marks (""):
  `lsarraymembergoals -filtervalue mdisk_id="1*"`

-filtervalue?
(Optional) Displays the valid filter attributes for the **-filtervalue** parameter:

- capacity_goal
- drive_id
- enclosure_id_goal
- estimated_completion_time
- mdisk_id
- mdisk_name
- member_id
- node_id_goal
- progress
- RPM_goal
- slot_id_goal
- tech_type_goal
- drive_class_id_goal

-delim *delimiter*
(Optional) By default, in a concise view all columns of data are space-separated, with the width of each column set to the maximum width of each item of data. In a detailed view, each item of data is an individual row, and if it displays headers, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. Enter `-delim :` on the command line, and the colon character (`:`) separates all items of data in a concise view (for example, the spacing of columns does not occur); in a detailed view, the data is separated from its header by the *delimiter* you specify.

-bytes
(Optional) Requests output of capacities in bytes (instead of rounded values).

mdisk_id
(Optional) The identity of the array MDisk.

mdisk_name
(Optional) The MDisk name that you provided.

Description

This command lists the spare goals for member drives of one or more array MDisks. Table 15 provides the potential output for this command.

Table 15. lsarraymembergoals output

Attribute	Values
member_id	Indicates the ID of the array member that represents the drive order in the RAID array.
drive_id	Indicates the ID of the drive for the member ID (it is blank if none are configured).
capacity_goal	Indicates the capacity goal for the array member (it is the same for all members in the array).

Table 15. lsarraymembergoals output (continued)

Attribute	Values
tech_type_goal	Indicates the technology goal for the array member: • tier0_flash • tier1_flash • tier_enterprise • tier_nearline
RPM_goal	Indicates the drive RPM characteristic that the array member must have (it is blank for flash drives).
enclosure_id_goal	Indicates the ID of the enclosure that must contain the array member (it is blank if no enclosure is selected).
slot_id_goal	Indicates the ID of the slot in the enclosure that must contain the array member.
node_id_goal	Indicates the ID of the node that must contain this array member.
enclosure_balance_goal	Indicates whether a member drive's enclosure must be connected to the same SAS port as the drive that set the array goals.
block_size_goal	Indicates the array member block size. The value is either 512 or 4096. This value is the same for all member drives in the array, and is the smallest value for the block size of one of the original drives or the set of drives in the array when it is set to be balanced.
drive_class_id_goal	Indicates the preferred drive class for this array member (the value is blank for nondistributed arrays).

An invocation example (a four-member RAID-10 SAS array that is split across chains)

You cannot use RAID-10 with distributed arrays.

```
lsarraymembergoals -delim : mdisk_2
```

The resulting output:

```
mdisk_id:mdisk_name:member_id:drive_id:capacity_goal:
tech_type_goal:RPM_goal:enclosure_id_goal:slot_id_goalenclosure_balance_goal:node_id_goal:block_size_goal:drive_class_id_go
2:mdisk2:0:0:68.4GB:tier0_flash:15000:1:1:no:512:0
2:mdisk2:1:17:68.4GB:tier0_flash:15000:1:2:no:512:0
2:mdisk2:2:1:68.4GB:tier0_flash:15000:14:1:no:512:2
2:mdisk2:3:18:68.4GB:tier0_flash:15000:14:2:no:512:2
```

An invocation example

```
lsarraymembergoals -filtervalue block_size_goal=4096
```

The resulting output:

```
mdisk_id  mdisk_name  member_id  drive_id  capacity_goal  tech_type_goal  RPM_goal  enclosure_id_goal  slot_id_goal  node_id_goal  enclosure_balance_goal  block_size_goal  drive_class_
4         r10_array   0          43        1.6TB          tier0_flash         10000     1                  21                          no                      4096  0
4         r10_array   1          44        1.6TB          tier0_flash         10000     1                  18                          no                      4096  0
4         r10_array   2          45        1.6TB          tier0_flash         10000     1                  20                          no                      40962
4         r10_array   3          46        1.6TB          tier0_flash         10000     2                   5                          no                      40962
```

```
lsarraymembergoals
```

The resulting output:

```
mdisk_id mdisk_name member_id drive_id capacity_goal tech_type_goal RPM_goal enclosure_id_goal slot_id_goal node_id_goal enclosure_b
0        r10_array  0         1        278.9GB       tier1_flash        15000    1                 2                        no
0        r10_array  1         10       278.9GB       tier1_flash        15000    1                 3                        no
0        r10_array  2         9        278.9GB       tier1_flash        15000    1                 4                        no
0        r10_array  3         0        278.9GB       tier1_flash        15000    1                 5                        no
0        r10_array  4         6        278.9GB       tier1_flash        15000    1                 6                        no
0        r10_array  5         7        278.9GB       tier1_flash        15000    1                 7                        no
0        r10_array  6         18       278.9GB       tier1_flash        15000    1                 8                        no
```

```
0       r10_array  7      21     278.9GB     tier1_flash     15000    1              9                   no
1       r0_array   0      15     278.9GB     tier1_flash     15000    1              10                  no
1       r0_array   1      22     278.9GB     tier1_flash     15000    1              11                  no
1       r0_array   2      13     278.9GB     tier1_flash     15000    1              12                  no
1       r0_array   3      5      278.9GB     tier1_flash     15000    1              13                  no
2       r1_array3  0      8      278.9GB     tier1_flash     15000    1              14                  no
2       r1_array3  1      4      278.9GB     tier1_flash     15000    1              15                  no
3       r1_array1  0      16     278.9GB     tier1_flash     15000    1              16                  no
3       r1_array1  1      12     278.9GB     tier1_flash     15000    1              17                  no
4       r1_array2  0      17     278.9GB     tier1_flash     15000    1              20                  no
4       r1_array2  1      19     278.9GB     tier1_flash     15000    1              19                  no
```

lsarraymemberprogress

Use the **lsarraymemberprogress** command to display array member background process status.

Syntax

```
►►── lsarraymemberprogress ── ─┬─────────┬─ ─┬───────────────────────────────┬─ ──────────►
                               └─ -nohdr ─┘   └─ -filtervalue ── attribute_value ─┘

►─┬─────────────────┬─ ─┬─────────────────────┬─ ─┬──────────────┬────────────────────────►◄
  └─ -filtervalue? ─┘   └─ -delim ──delimiter─┘   ├─mdisk_id─────┤
                                                  └─mdisk_name───┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards:

- The wildcard character is the asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, enclose the filter entry within double quotation marks (""):
  `lsarraymemberprogress -filtervalue mdisk_id="1*"`

-filtervalue?
(Optional) Displays the valid filter attributes for the **-filtervalue** parameter:

- estimated_completion_time
- drive_id
- mdisk_id
- mdisk_name
- member_id
- new_drive_id
- progress
- task

-delim *delimiter*

(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

mdisk_id

(Optional) The identity of the array MDisk.

mdisk_name

(Optional) The MDisk name that you provided.

Description

This command displays array member background process status. Exchange cannot start on a rebuilding member because both component rebuild and exchange are shown in the same view. Table 16 provides the potential output for this command.

Table 16. lsarraymemberprogress output

Attribute	Value
member_id	Indicates the array member index.
drive_id	Indicates the ID of the drive.
task	Indicates the identity of task that is being performed by the array member: • `rebuild` indicates that the array is recovering all the data on the component (after it was removed) • `exchange` indicates that the component is copying data to another drive • `resync` indicates that this member is unsynchronized and is performing write operations that were completed early • `copyback` indicates that this member is copying data to an array member that recently became active. **Note:** This value applies to distributed arrays. **Note:** For example, if the drive fails the array is rebuilt. If the drive does not fail or is replaced, a copyback occurs to write data back to the array member.
new_drive_id	The identity of drive that is being exchanged.
progress	Indicates the task percentage completion.
estimated_completion_time	Indicates the expected task completion time in the format `YYMMDDHHMMSS`. It is blank if completion time is unknown.

A concise invocation example

```
lsarraymemberprogress –delim :
```

The resulting output:

```
mdisk_id:mdisk_name:member_id:drive_id:task:new_drive_id:progress:estimated_completion_time
0:mdisk0:2:3:rebuild::50:070301120000
1:mdisk1:0:5:rebuild::51:070301130000
2:mdisk2:4:1:exchange:12:32:070301153500
2:mdisk2:5:16:exchange:13:0:
2:mdisk2:5:17:exchange:14:0:
```

An MDisk qualified concise example

```
lsarraymemberprogress mdisk_2
```

The resulting output:

```
mdisk_id:mdisk_name:member_id:drive_id:task:new_drive_id:progress:estimated_completion_time
2:mdisk2:4:1:exchange:12:32:070301153500
2:mdisk2:5:16:exchange:13:0:
2:mdisk2:5:17:exchange:14:0:
```

An invocation example

```
lsarraymemberprogress
```

The resulting output:

```
mdisk_id mdisk_name member_id drive_id task     new_drive_id progress estimated_completion_time
3        mdisk3     5         1        resync                95       121203193637
3        mdisk3     6         2        rebuild               0        121203234321
3        mdisk3     7         3        exchange 18           0        121204033229
```

An invocation example

```
lsarraymemberprogress
```

The resulting output:

```
mdisk_id mdisk_name member_id drive_id task     new_drive_id progress estimated_completion_time
3        mdisk3     4         7        copyback              10       150710165446
```

lsarrayrecommendation

Use the **lsarrayrecommendation** command to view a recommended configuration for the specified drive class and number of drives.

Syntax

```
►►— lsarrayrecommendation — — -driveclass —drive_class_id_list— ————————————————————————————————►

►— -drivecount —drive_count_list— ┬————————————┬ ┬———————————————————┬ ——————————————————————————►
                                  └ -nohdr ————┘ └ -filtervalue? —————┘

►┬————————————————————————————————┬ ┬————————————————————————┬ ┬— mdiskgrp_id ———┬————————————►◄
 └ -filtervalue — attribute=value ┘ └ -delim — delimiter ————┘ └— mdiskgrp_name ——┘
```

Parameters

-driveclass *drive_class_id_list*
(Required) Specifies the drive class, or classes, for which the array recommendation is made. You must specify at least 1 *drive_class_id_list* value. You can specify a total of 32 *drive_class_id_list* values on a single command; however, you must separate each value with a colon character (**:**).

-drivecount *drive_count_list*
(Required) Specifies the number of drives for which to make recommendation. You must specify at

least 1 *drive_count* value. You can specify a total of 32 *drive_count* values on a single command; however, you must separate each value with a colon character (`:`).

Remember: Each `drive_count` value must be a numerical value between 2 and 128 (only redundant arrays are considered).

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filter attributes that match the specified values; see **-filtervalue?** for the supported attributes.

Note: Some filters allow the use of a wildcard; the following rules apply when using a wildcard character:

- The wildcard character is an asterisk (*).
- The command can contain a maximum of one wildcard, which must be the first or last character in the string.
- You must enclose the filter entry within double quotation marks ("").

-filtervalue?
(Optional) Includes all of the valid filter attributes in the report. The following filter attributes are valid for the **lsarrayrecommendation** command:

- raid_level
- distributed

Any parameters that are specified with the **-filtervalue?** parameter are ignored.

-delim *delimiter*
(Optional) By default, in a concise view all columns of data are space-separated, with the width of each column set to the maximum width of each item of data. In a detailed view, each item of data is an individual row, and if you display headers, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. Enter `-delim :` on the command line. The colon character (`:`) separates all items of data in a concise view (for example, the spacing of columns does not occur). In a detailed view, the specified *delimiter* separates the data from its header.

mdiskgrp_id
mdiskgrp_name
(Required) The ID or name of the pool for which to make the recommendation.

Description

This command displays the system-recommended array configuration for a specific drive class and number of drives.

Encrypted pools can be recommended or used if the specified storage pool is encrypted.

Table 17 provides the attribute values that can be displayed as output view data.

Table 17. lsarrayrecommendation output

Attribute	Possible Values
mdiskgrp_id	Indicates the MDisk group ID

Table 17. lsarrayrecommendation output (continued)

Attribute	**Possible Values**
mdiskgrp_name	Indicates the MDisk group name
drive_class_id	Indicates the drive class ID for the recommendation.
raid_level	Indicates the recommended RAID level.
distributed	Indicates whether the array is a distributed array recommendation.
min_stripe_width	Indicates the stripe width minimum.
max_stripe_width	Indicates the stripe width maximum.
stripe_width	Indicates the recommended stripe size for this storage pool.
rebuild_areas	Indicates the recommended number of rebuild areas.
strip_size	Indicates the recommended strip size for this storage pool.
drive_count	Indicates the number of drives to configure per array.
array_count	Indicates the number of arrays that can be created at the recommended drive count.
capacity	Indicates the expected usable capacity for virtualization for the array or arrays.

Note: The default recommendation for each drive class is the first row that is listed in the output for that drive class. All other rows for each drive class are not sorted by order of recommendation.

An invocation example

Making a recommendation for more rebuild areas and larger stripe width because there are other existing arrays in the MDisk group:

```
lsarrayrecommendation -driveclass 2 -drivecount 60 dist_pool
```

The detailed resulting output:

```
mdisk_grp_id mdisk_grp_name drive_class_id raid_level distributed min_stripe_width max_stripe_width stripe_width rebuild_areas st
0            dist_pool      2              raid6      yes         5                16               12           3             25
```

An invocation example

Making a recommendation for a new (empty) storage pool and with a row for each raid level:

```
lsarrayrecommendation -driveclass 2 -drivecount 80 mdiskgrp0
```

The detailed resulting output:

```
mdisk_grp_id mdisk_grp_name drive_class_id raid_level distributed min_stripe_width max_stripe_width stripe_width rebuild_areas strip_size drive_count array_count capacity
 0            mdiskgrp0      2              raid5      yes         3                16               6            2             256        40          2           410.1TB
 0            mdiskgrp0      2              raid6      yes         5                16               12           3             256        40          2           400.1TB
 0            mdiskgrp0      2              raid5      yes         2                16               8            2             256        40          2           380.1TB
 0            mdiskgrp0      2              raid1      no          2                16               2                          256        10          8           410.1TB
 0            mdiskgrp0      2              raid5      no          3                16               10                         256        10          7           410.1TB
 0            mdiskgrp0      2              raid6      no          5                16               10                         256        10          7           400.1TB
 0            mdiskgrp0      2              raid5      no          2                16               8                          256        8           9           380.1TB
```

An invocation example

Making a recommendation for multiple drive classes for a new (empty) storage pool:

```
lsarrayrecommendation -driveclass 3:5 -drivecount 80:24 mdiskgrp0
```

The detailed resulting output:

```
mdisk_grp_id mdisk_grp_name drive_class_id raid_level distributed min_stripe_width max_stripe_width stripe_width rebuild_areas strip_size drive_count array_count capacity
  0            mdiskgrp0      3              raid5      yes         3                16               6            2             256       40          2           500.4TB
  0            mdiskgrp0      3              raid6      yes         5                16               12           3             256       40          2           480.4TB
  0            mdiskgrp0      3              raid5      yes         2                16               8            2             256       40          2           450.4TB
  0            mdiskgrp0      3              raid1      no          2                16               2                          256       10          8           400.6TB
  0            mdiskgrp0      3              raid5      no          3                16               10                         256       10          7           500.6TB
  0            mdiskgrp0      3              raid6      no          5                16               10                         256       10          7           480.6TB
  0            mdiskgrp0      3              raid5      no          2                16               8                          256       8           9           450.6TB
  0            mdiskgrp0      5              raid5      yes         3                16               6            2             256       12          2           200.7TB
  0            mdiskgrp0      5              raid6      yes         5                16               12           3             256       12          2           180.7TB
  0            mdiskgrp0      5              raid5      yes         2                16               8            2             256       8           3           150.7TB
  0            mdiskgrp0      5              raid1      no          2                16               2                          256       8           3           100.2TB
  0            mdiskgrp0      5              raid5      no          3                16               12                         256       12          2           200.2TB
  0            mdiskgrp0      5              raid6      no          5                16               12                         256       12          2           180.2TB
  0            mdiskgrp0      5              raid5      no          2                16               8                          256       8           3           150.2TB
```

lsarraysyncprogress

Use the **lsarraysyncprogress** command to display how synchronized a RAID array is.

Syntax

```
►►─lsarraysyncprogress──┬──────────┬──┬─────────────────────────────────────┬──────────────────────►
                        └─ -nohdr ─┘  └─ -filtervalue ── attribute_value ─┘

►──┬───────────────────┬──┬─────────────────────────┬──┬──────────────┬──────────────────────────►◄
   └─ -filtervalue? ─┘  └─ -delim ──delimiter─┘  ├─mdisk_id───┤
                                                 └─mdisk_name─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards:

- The wildcard character is the asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, enclose the filter entry within double quotation marks (""): `lsarraysyncprogress -filtervalue mdisk_id="1*"`

-filtervalue?
(Optional) Displays the valid filter attributes for the **-filtervalue** parameter:

- estimated_completion_time
- mdisk_id
- mdisk_name
- progress

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (:) separates all items of data in a

concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

mdisk_id
(Optional) The ID of the MDisk you want to view.

mdisk_name
(Optional) The user-defined name of the MDisk you want to view.

Description

This command shows you how synchronized a RAID array is. It includes internal activity that is working toward a fully synchronized array. Table 18 provides the potential output.

Table 18. lsarraysyncprogress output

Attribute	Value
progress	The percentage of the array that is synchronized.
estimated_completion_time	The expected synchronization completion time (`YYMMDDHHMMSS`; blank if completion time unknown).

A concise invocation example

```
lsarraysyncprogress –delim :
```

The resulting output:

```
mdisk_id:mdisk_name:progress:estimated_completion_time
0:mdisk0:50:070301120000
1:mdisk1:51:070301130000
2:mdisk2:32:070301153500
```

A concise view (qualified with mdisk id for mdisk2) invocation example

```
lsarraysyncprogress –delim : mdisk2
```

The resulting output:

```
mdisk_id:mdisk_name:progress:estimated_completion_time
2:mdisk2:32:070301153500
```

A concise view (qualified with mdisk id for in sync mdisk10) invocation example

```
lsarraysyncprogress –delim : mdisk_10
```

The resulting output:

```
mdisk_id:mdisk_name:progress:estimated_completion_time
0:mdisk10:100:
```

lspotentialarraysize

Use the **lspotentialarraysize** command to display the size of a potential array for a specified drive count, drive class, and RAID level in the specified MDisk group.

Syntax

```
►►── lspotentialarraysize ── ┬──────────┬── ┬────────────────────┬── ──────────────────────────►
                             └─ -nohdr ─┘   └─ -delim ── delimiter ─┘
```

```
►─ -drivecount ── 3- 128 ── ── -driveclass ── drive_class_id ── ── -level ──┬─raid1──┬─ ──────────────►
                                                                             ├─raid5──┤
                                                                             ├─raid6──┤
                                                                             └─raid10─┘

►─ -stripewidth ── 2 - 16 ── ──────────────────────────────── ─────────────────────────── ──────────────►
                              └─ -rebuildareas ──┬─ 1 ─┬─┘     └─ -strip ──┬─ 128 ─┬─┘
                                                 ├─ 2 ─┤                   └─ 256 ─┘
                                                 ├─ 3 ─┤
                                                 └─ 4 ─┘

►──┬─ mdiskgrp_id ───┬──────────────────────────────────────────────────────────────────────────────►◄
   └─ mdiskgrp_name ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default, in a concise view all columns of data are space-separated, with the width of each column set to the maximum width of each item of data. In a detailed view, each item of data is an individual row, and if you display headers, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. Enter -delim : on the command line; the colon character (:) separates all items of data in a concise view (for example, the spacing of columns does not occur). In a detailed view, the specified *delimiter* separates the data from its header.

-drivecount *3 - 128*
(Required) Specifies the number of the drives. The value must be a number from 3 - 128.

-driveclass *drive_class_id*
(Required) Specifies the drive class. The *driveclass_id* value must be a number.

-level *raid1* | *raid5* | *raid6* | *raid10*
(Required) Specifies one of the following RAID levels for the array that is being created. The values are:

- *raid1*
- *raid5*
- *raid6*
- *raid10*

-stripewidth *2 - 16*
(Required) Indicates the width of a single unit of redundancy within a distributed set of drives. The value must be:

- RAID-1: 2 - 16
- RAID-5: 3 - 16
- RAID-6: 5 - 16
- RAID-10: 2, 4, 6, 8, 10, 12, 14, 16 (You cannot use RAID-10 with distributed arrays.)

-rebuildareas *1* | *2* | *3* | *4*
(Optional) Specifies the number of rebuild areas in the array. This value must be 1 - 4 (inclusive) for RAID-5 and RAID-6 arrays.

Note: This parameter is only applicable for distributed arrays.

-strip *128* | *256*
(Optional) Specifies sets the strip size in KiB for the array that is being configured. The values are 128 or 256.

mdiskgrp_id | *mdiskgrp_name*
(Required) Indicates the MDisk array ID or name.

Description

This command displays the size of a potential array for a specified drive count, class, and RAID level.

You can use this command to define potential sizes for nondistributed and distributed arrays. (Distributed array descriptions are triggered by using **-rebuildareas**.) This command assists with the configuration options that are provided during array creation, and estimates the array capacity if it were to be configured

Table 19 provides the attribute values that can be displayed as output view data.

Table 19. lspotentialarraysize output

Attribute	**Possible Values**
capacity	Indicates the expected usable capacity for virtualization for the array or arrays.

An invocation example that uses a small drive count for a distributed array

```
lspotentialarraysize -driveclass 4 -drivecount 40 -level raid5 -stripewidth 6 -rebuildareas 2 mdiskgrp1
```

The detailed resulting output:

```
capacity
115.2TB
```

An invocation example that uses a different class and fewer rebuild areas

```
lspotentialarraysize -driveclass 4 -drivecount 100 -level raid5 -stripewidth 8 -strip 128 -rebuildareas 1 mdiskgrp1
```

The detailed resulting output:

```
capacity
172.4TB
```

An invocation example that uses the same class and no rebuild areas

```
lspotentialarraysize -driveclass 4 -drivecount 100 -level raid5 -stripewidth 8 -strip 128 1
```

The detailed resulting output:

```
capacity
184.3TB
```

mkarray

Use the **mkarray** command to create an MDisk array and add it to a storage pool. This command applies to nondistributed arrays. (Use the **mkdistributedarray** command to create distributed arrays).

Syntax

```
►►─mkarray─ -drive ─drive_id_list──┬───────────────────┬─┬────────────────────────────────┬──────►
                                   └─ -strip ──┬128─┬──┘ └─ -sparegoal ──0-(MAX_DRIVES-1)──┘
                                               └256─┘
```

```
►──┬────────────────────────┬──┬──────────────────────────────────────────────┬──┬───────────────────────────┬──────►
   └─ -name ── new_name_arg─┘  └─ -slowwritepriority ──┬─latency────┬─────────┘  └─ -encrypt ──┬─yes─┬──────┘
                                                       └─redundancy─┘                         └─no──┘

►──┬─mdiskgrp_id───┬──────────────────────────────────────────────────────────────────────────────────────►◄
   └─mdiskgrp_name─┘
```

Parameters

-level
(Required) Sets the RAID level for the array MDisk being created.

The following requirements apply for RAID levels:

- RAID-0: Stripes data across all members, provides no redundancy.
- RAID-1: Mirrored pair of drives, allows reading from either drive. Can tolerate either drive failing.
- RAID-5: These arrays stripe data over the member drives with one parity strip on every stripe and can tolerate no more than one member drive failure.
- RAID-6: These arrays stripe data over the member drives with two parity strips on every stripe and can tolerate any two concurrent member drive failures.
- RAID-10: These arrays are in a set of up to eight mirrored pairs with the data striped across mirrors. They can tolerate the failure of one drive in each mirror and they allow reading from both drives in a mirror. (You cannot use RAID-10 with distributed arrays).

Restriction: RAID-5 and RAID-6 are for enclosure-based systems only.

-drive *drive_id_list*
(Optional) Identifies the drive or drives to use as members of the RAID array.

For RAID-1 and RAID-10 arrays, drives are specified as a sequence of mirrored drive pairs. For example, if an array is created with `-drive a:b:c:d`, drive *b* contains the mirror copy of drive *a*, and drive *d* contains the mirror copy of drive *c*. (You cannot use RAID-10 with distributed arrays).

This list shows how many member drives are allowed in each supported RAID type:

- RAID-0: Allows one-member to eight-member drives.

  Note: Internal drives must be in the same node.
- RAID-1: Allows two-member drives.
- RAID-5, which is for enclosure-based systems only: Allows three-member to 16-member drives.
- RAID-6, which is for enclosure-based systems only: Allows five-member to 16-member drives.
- RAID-10: Allows drives with:
  - Two members
  - Four members
  - Six members
  - Eight members
  - Ten members
  - Twelve members
  - Fourteen members
  - Sixteen members

  Each pair of drives must contain a drive from a node in the I/O group and a drive from the other node. (You cannot use RAID-10 with distributed arrays.)

-strip *128* **|** *256*
(Optional) Sets strip size (in KB) for the array MDisk being created. The default is 256 KB.

-sparegoal *0-(MAX_DRIVES-1)*
(Optional) Sets the number of spares that this array's members must be protected by. The default is 1 (except for RAID-0 arrays, which have a default of 0).

-name *new_name_arg*
(Optional) Specifies the name to which you want to apply the array MDisk.

-slowwritepriority latency | redundancy
(Optional) Controls array ability to complete write operations that take too long, even if it temporarily compromises redundancy.

The value can be either latency or redundancy:

- latency implies that the feature is enabled for normal I/O operations
- redundancy implies that the feature is not enabled for normal I/O operations

The default value is latency mode for existing arrays, unless the array is RAID-0 (in which case redundancy mode is required).

Important: Do not change the mode of a RAID-0 array.

-encrypt *yes* | *no*
(Optional) Specifies the array to encrypt. The values are *yes* and *no*.

This parameter defaults to *yes* when **lsencryption** has its status set to enabled and all nodes in the I/O group that the array is being defined on are encryption-capable.

Note: The value can be *yes* only if encryption is enabled on the array's I/O group.

mdiskgrp_id | *mdiskgrp_name*
(Required) Identifies the storage pool (by name or ID) to which you want to add the created array MDisk.

Description

This command creates an array MDisk RAID array and adds it to a storage pool. Although the array tier is automatically determined, you can change it later using the **chmdisk** command.

An array MDisk being added to a storage pool that is used for active-active relationships must match other MDisks in the storage pool.

Remember: This command cannot be used to add an array to a child pool.

If the raid_level is RAID-1 or RAID-10, and the drive list contains drives that do not share a SAS port connection chain, the array attempts to continue to maintain the location balance between the mirrored pairs. (You cannot use RAID-10 with distributed arrays.) Configuration changes indicate that a member drive might not be goal-balanced depending on its current chain. This is relative to both the drive that created the array member goals and the current chain of the mirror partner.

If the MDisk group has an encryption key, the array must be encrypted.

An invocation example (to create arrays)

```
mkarray -level raid0 -drive 0:1:2:3 raid0grp
```

The resulting output:

```
MDisk, id [0], successfully created
```

An invocation example (to create fully redundant arrays)

```
mkarray -level raid1 -drive 4:5 -strip 128 mdiskgrp_4
```

The resulting output:

```
MDisk, id [1], successfully created
```

An invocation example for creating an unencrypted array on encrypted hardware

```
mkarray -level raid10 -drives 0:1:2:3:4:5 -encrypt no 0
```

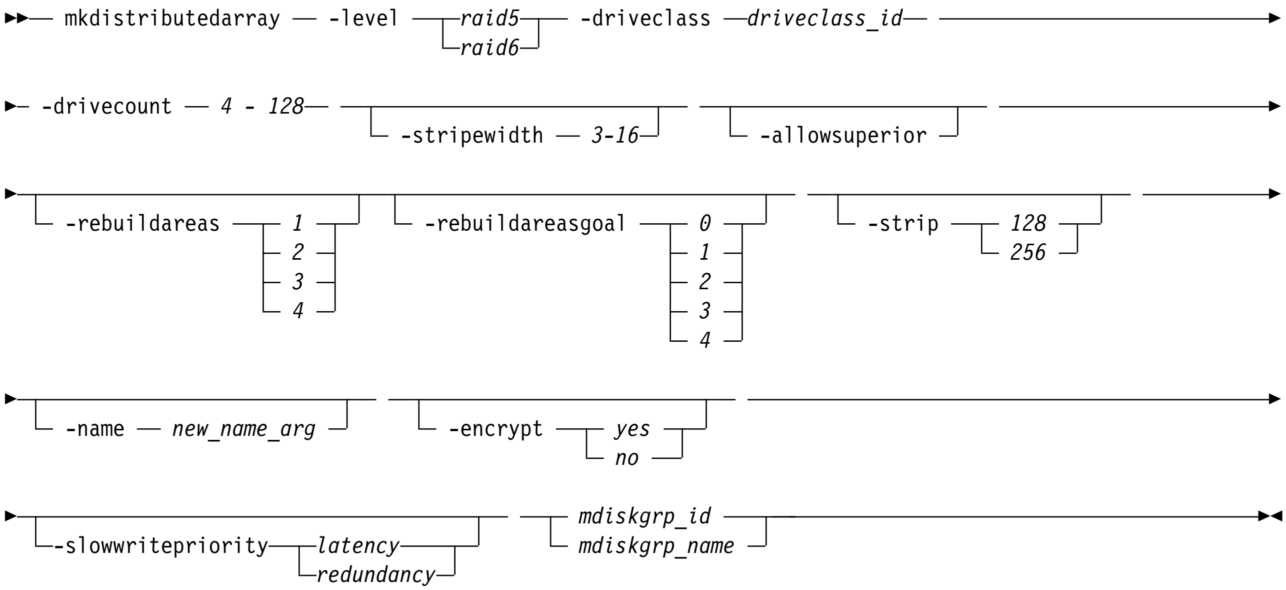
The resulting output:

```
MDisk, id [1], successfully created
```

mkdistributedarray

Use the **mkdistributedarray** command to create a distributed array and add it to a storage pool. (Use the **mkarray** command to create nondistributed arrays).

Syntax



Parameters

-level *raid5* | *raid6*
(Required) Specifies the RAID level for the array that is being created. The values are:

- *raid5*
- *raid6*

-driveclass *driveclass_id*
(Required) Specifies the class that is being used to create the array. The *driveclass_id* must be a numeric value (specified with the **lsdriveclass** command).

-drivecount *4 - 128*
(Required) Specifies the number of drives to use for the array. The minimum drive count for:

- RAID-5: 4
- RAID-6: 6

-stripewidth *3-16*

(Optional) Indicates the width of a single unit of redundancy within a distributed set of drives. The value must be:

- RAID-5: 3 - 16
- RAID-6: 5 - 16

The default width for RAID-6 is 12 and the default width for RAID-5 is 10. The width plus the number of rebuild areas must be less than or equal to the drive count.

-allowsuperior

(Optional) Specifies that you can use drives that are not an exact match to the drive class used when creating the array (such as drives that use different capacity or technology). The system attempts to select the closest match to the class when satisfying the drive count. You can select higher capacity members of the same technology type before you select higher technology members.

Note: For a drive A to be considered superior to drive B, these situations must be true:

1. Drives A and B are use=candidate
2. Drives A and B are in the same I/O group.
3. Drive A's speed (RPM) is equal to or greater than drive B's. Solid-state drives (SSDs) are higher speed than all hard disk drives (HDDs).
4. Drive A's capacity is equal to or greater than drive B's.
5. Drive A has a block size that is smaller than or equal to drive B.

-rebuildareas *1* | *2* | *3* | *4*

(Optional) Specifies the reserved capacity that is distributed across all drives available to an array. This capacity restores data after a drive failure. The values are:

- 1
- 2
- 3
- 4

The value is 1 - 4 (inclusive) for RAID-5 and RAID-6 arrays.

The default number of rebuild areas increases as the drive count increases.

Note: The number of rebuild areas plus the stripe width must be less than or equal to the total drive count.

-rebuildareasgoal *0* | *1* | *2* | *3* | *4*

(Optional) Specifies the number of rebuild areas that the array can target to keep available. If the number available in the array falls below this number, a system alert is raised.

Note: The goal value should not exceed the number of rebuild areas that are specified for the array. The values are:

- 0
- 1
- 2
- 3
- 4

-strip *128* | *256*

(Optional) Specifies the strip size in KiB for the array that is being configured. The values are 128 or 256.

Note: This command fails if 128 is specified and the size of the candidate drives is greater than 4 TB.

-name *new_name_arg*
(Optional) Specifies the name of the array.

-encrypt *yes* | *no*
(Optional) Specifies the array to encrypt. The values are `yes` and `no`. This parameter defaults to `yes` when **lsencryption** has its status set to `enabled` and all nodes in the I/O group that the array is being defined on are encryption-capable.

Note: The value can be `yes` only if encryption is enabled on the array's I/O group.
If you specify `-encrypt yes` when the I/O group does not support encryption, the command fails.

-slowwritepriority latency | redundancy
(Optional) Controls array ability to complete write operations that take too long, even if it temporarily compromises redundancy.

The value can be either `latency` or `redundancy`:

- `latency` implies that the feature is enabled for normal I/O operations
- `redundancy` implies that the feature is not enabled for normal I/O operations

The default value is `latency` mode for existing arrays).

Important: An array can cause member drives to become unsynchronized (to preserve response time) if the value is `latency`. If the value is `redundancy`, the array cannot cause member drives to become unsynchronized (to preserve time) and I/O performance is impacted.

mdiskgrp_id | *mdiskgrp_name*
(Required) Indicates the MDisk array ID or name.

Description

This command creates distributed arrays.

Remember: You cannot create an unencrypted array to add to an encrypted storage pool.
Each distributed array occupies 16 slots, which start at an MDisk ID that is divisible by 16. See the **lsmdisk** command for more information.

An invocation example to create an array that uses 40 drives of class 3 with 3 rebuild areas

```
 mkdistributedarray -level raid6 -driveclass 3 -drivecount 40 -stripewidth 10 -rebuildareas 3 mdiskgrp5
```

The detailed resulting output:

```
MDisk, id [16], sucessfully created
```

An invocation example to create an array with a drive class

```
 mkdistributedarray -level raid5 -driveclass 0 -drivecount 56 -stripewidth 8 -allowsuperior mdiskgrp2
```

The detailed resulting output:

```
MDisk, id [32], sucessfully created
```

An invocation example to create an array with maximum rebuild areas that logs an error on using the second rebuild area

```
mkdistributedarray -level raid5 -driveclass 5 -drivecount 60 -rebuildareas 4 -rebuildareasgoal 3 mdiskgrp2
```

The detailed resulting output:

```
MDisk, id [16], sucessfully created
```

An invocation example to create an array that might affect I/O performance

```
mkdistributedarray -driveclass 10 -slowwritepriority redundancy 0
```

The detailed resulting output:

```
MDisk, id [32], sucessfully created
```

An invocation example to make an encrypted distributed array that uses 40 drives of class 3 with three rebuild areas

```
mkdistributedarray -level raid6 -driveclass 3 -drivecount 40 -stripewidth 10 -rebuildareas 3 -encrypt yes mdiskgrp5
```

The detailed resulting output:

```
MDisk, id [16], sucessfully created
```

recoverarray

Use the **recoverarray** command to recover a specific corrupt array in a dead domain scenario.

Syntax

```
►►──recoverarray──┬─mdisk_id───┬──────────────────────────────────────────────────────►◄
                  └─mdisk_name─┘
```

Parameters

mdisk_id
(Optional) Identifies (by ID) the specific array to recover.

mdisk_name
(Optional) Identifies (by user-assigned name) the specific array to recover.

Description

This command recovers a specific corrupt array. An array has metadata representing ongoing or pending platform writes, which are lost when the domain nodes are lost.

An invocation example

```
recoverarray mdisk_1
```

The resulting output:

```
There is no output if the command is successful.
```

recoverarraybycluster (Discontinued)

Attention: The **recoverarraybycluster** command has been discontinued. Use the **recoverarraybysystem** command instead.

recoverarraybysystem

Use the **recoverarraybysystem** command to recover corrupt arrays in a dead domain scenario.

Syntax

```
►►──recoverarraybysystem──────────────────────────────────────────────────────────────►◄
```

Parameters

None.

Description

Use the **recoverarraybysystem** command to recover corrupt arrays in a dead domain scenario.

An invocation example

```
recoverarraybysystem
```

The resulting output:

There is no output if the command is successful.

rmarray

Use the **rmarray** command to remove an array MDisk from the configuration.

Syntax

```
►►──rmarray── -mdisk ──┬─mdisk_id_list───┬──┬──────────┬──┬─mdiskgrp_id───┬──────────────────────►◄
                       └─mdisk_name_list─┘  └─ -force ─┘  └─mdiskgrp_name─┘
```

Parameters

-mdisk *mdisk_id_list* | *mdisk_name_list*
(Required) Identifies the array MDisk or a colon-delimited list of MDisks to remove from the storage pool.

-force
(Optional) Forces a remove when the MDisk has allocated extents by migrating the used extents to free extents in the storage pool.

mdiskgrp_id | *mdiskgrp_name*
(Required) Identifies (by name or ID) the storage pool to remove the created array MDisk from.

Description

This command removes an array MDisk from the configuration. Each array is divided into candidate drives.

Remember: This command cannot be used to remove an array MDisk from a child pool.

An invocation example

```
rmarray -mdisk 6 mdiskgrp_10
```

The resulting output:

```
No feedback
```

Chapter 5. Audit log commands

Use the audit log commands to track command specifications and related data. An audit log keeps track of action commands that are issued through a Secure Shell (SSH) session or through the management GUI.

The audit log entries provide the following information:

- Identity of the user who issued the action command
- The name of the actionable command
- The timestamp of when the actionable command was issued on the configuration node
- The parameters which were issued with the actionable command

The following commands are not documented in the audit log:

- **dumpconfig**
- **cpdumps**
- **finderr**
- **dumperrlog**

The following items are also not documented in the audit log:

- Commands that fail are not logged
- A result code of 0 (success) or 1 (success in progress) is not logged
- Result object ID of node type (for the **addnode** command) is not logged
- Views are not logged

catauditlog

Use the **catauditlog** command to display the in-memory contents of the audit log.

Syntax

```
►►── catauditlog ──┬─────────┬── ┬──────────────────────┬── ──────────────────────────────►
                   └─ -nohdr ┘   └─ -delim ── delimiter ┘

►──┬──────────────────────────────────────────┬── ─────────────────────────────────────►◄
   └─ -first ── number_of_entries_to_return ──┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum possible width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a one-byte character. If you enter -delim : on the command line, the colon character (:) separates all

items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-first *number_of_entries_to_return*
(Optional) Specifies the number of most recent entries to display.

Description

This command lists a specified number of the most recently audited commands.

Use this command to display the in-memory audit log. Use the **dumpauditlog** command to manually dump the contents of the in-memory audit log to a file on the current configuration node and clear the contents of the in-memory audit log

The in-memory portion of the audit log holds approximately 1 MB of audit information. Depending on the command text size and the number of parameters, this equals 1 MB of records or approximately 6000 commands.

Once the in-memory audit log reaches maximum capacity, the log is written to a local file on the configuration node in the /dumps/audit directory. The **catauditlog** command only displays the in-memory part of the audit log; the on-disk part of the audit log is in readable text format and does not require any special command to decode it.

The in-memory log entries are reset and cleared automatically, ready to accumulate new commands. The on-disk portion of the audit log can then be analyzed at a later date.

The **lsdumps** command with **-prefix** parameter (and the /dumps/audit file) can be used to list the files on the disk.

As commands are executed, they are recorded in the in-memory audit log. When the in-memory audit log becomes full, it is automatically dumped to an audit log file and the in-memory audit log is cleared.

An invocation example

This example lists the five most recent audit log entries.

```
catauditlog -delim : -first 5
```

The resulting output:

```
audit_seq_no timestamp    cluster_user challenge  source_panel target_panel ssh_ip_address result res_obj_id action_cmd
0            160313152255 superuser               7830619-2    7830619-2                   0      0          satask restart
1            160313152303 superuser               01-2         01-1         9.174.187.11   0      0          satask chnodel
2            160313152312 superuser               01-1         01-2         9.174.187.11   0      0          satask chnodel
3            160313152314 superuser               01-1         01-1         9.174.187.11   0      0          satask chnodel
4            160313152316 superuser                                         9.174.187.11   0      0          svctask chencl
5            160313152349 superuser                                         9.174.187.11   0      0          svctask mkmdis
6            160313152352 superuser                                         9.174.187.11   0      0          svctask mkarra
```

dumpauditlog

Use the **dumpauditlog** command to reset or clear the contents of the in-memory audit log. The contents of the audit log are sent to a file in the /dumps/audit directory on the current configuration node.

Syntax

```
►►— dumpauditlog — ————————————————————————————————————————————————►◄
```

Parameters

There are no parameters.

Description

This command dumps the contents of the audit log to a file on the current configuration node in a clustered system (system). It also clears the contents of the audit log. This command is logged as the first entry in the new audit log.

Use this command to manually dump the contents of the in-memory audit log to a file on the current configuration node and clear the contents of the in-memory audit log. Use the **catauditlog** command to display the in-memory audit log.

Audit log dumps are automatically maintained in the `/dumps/audit` directory. The local file system space is used by audit log dumps and is limited to 200 MB on any node in the system. The space limit is maintained automatically by deleting the minimum number of old audit log dump files so that the `/dumps/audit` directory space is reduced below 200 MB. This deletion occurs once per day on every node in the system. The oldest audit log dump files are considered to be the ones with the lowest audit log sequence number. Also, audit log dump files with a system ID number that does not match the current one are considered to be older than files that match the system ID, regardless of sequence number.

Other than by running dumps (or copying dump files among nodes), you cannot alter the contents of the audit directory. Each dump file name is generated automatically in the following format:

`auditlog_`*`firstseq_lastseq_timestamp_clusterid`*

where

- *firstseq* is the audit log sequence number of the first entry in the log
- *lastseq* is the audit sequence number of the last entry in the log
- *timestamp* is the timestamp of the last entry in the audit log that is being dumped
- *clusterid* is the clustered system ID at the time that the dump was created
- *challenge* allows the `sra_privileged` user to determine who issued a particular command
- *source_panel* is the source panel ID in the audit log that is being dumped
- *target_panel* indicates the target panel ID in the audit log that is being dumped

The audit log dump files names cannot be changed.

The audit log entries in the dump files contain the same information as displayed by the **catauditlog** command; however, the **dumpauditlog** command displays the information with one field per line. The **lsdumps** command displays a list of the audit log dumps that are available on the nodes in the clustered system.

A sample audit log entry:

```
Auditlog Entry 0
       Sequence Num    : 0
       Timestamp       : Sun Mar 13 15:22:55 2016
                       : Epoch + 1457882575
       Cluster User    : superuser
       Challenge       :
       SSH IP Address  :
       Result Code     : 0
       Result Obj ID   : 0
       Action Cmd      : satask restartservice -service tomcat
       Source_Panel    : 7830619-2
       Target_Panel    : 7830619-2
```

An invocation example

```
dumpauditlog
```

The resulting output:

```
No feedback
```

lsauditlogdumps (Deprecated)

Attention: The **lsauditlogdumps** command is deprecated. Use the **lsdumps** command to display a list of files in a particular dumps directory.

Chapter 6. Backup and restore commands

Use the backup and restore commands to back up and restore configuration information about the system.

svcconfig

Use the **svcconfig** command help option to obtain summary information about the syntax of the **svcconfig** command and actions. You can enter this command any time after a clustered system (system) is created.

Syntax

```
►►─ svcconfig ── ── backup ── ── ─────────────── ───────────────────────────────────────────────►◄
                                  ├─ -q ─────┤    └─ -v ──┬─ off ─┬─┘
                                  └─ -quiet ─┘            └─ on ──┘

►►─ svcconfig ── ── clear ── ─────────── ─────────────── ──────────────────────────────────────►◄
                              └─ -all ─┘   ├─ -q ─────┤    └─ -v ──┬─ on ──┬─┘
                                           └─ -quiet ─┘            └─ off ─┘

►►─ svcconfig ── ── restore ── ─────────────── ─────────────── ──────────────────────────────────►
                                ├─ -f ─────┤    ├─ -q ─────┤    ├─ -prepare ──┬────────────┬─┤
                                └─ -force ─┘    └─ -quiet ─┘    │             ├─ -fmt ─────┤ │
                                                                │             └─ -fmtdisk ─┘ │
                                                                ├─ -execute ─────────────────┤
                                                                ├─ -fmt ─────────────────────┤
                                                                └─ -fmtdisk ─────────────────┘

►─ ───────────────────────── ─────────────────────────────────────────────────────────────────►◄
    └─ -v ──┬─ off ─┬─┘
            └─ on ──┘

►►─ svcconfig ── ── -ver ────────────────────────────────────────────────────────────────────►◄
```

Parameters

backup
(Optional) Saves the current clustered system (system) configuration in the /tmp directory.

-quiet
(Optional) Suppresses standard output (STDOUT) messages from the console.

clear
(Optional) Erases the files in the /tmp directory.

-all
(Optional) Erases all configuration files.

-f | force
(Optional) Forces continued processing where possible.

-q | quiet
(Optional) Suppresses console output (STDOUT).

restore
(Optional) Checks the current configuration against the backup configuration in the /tmp directory.

-prepare *-fmt | fmtdisk*
(Optional) Verifies the current configuration against the information in svc.config.backup.xml; then prepares commands for processing in svc.config.restore.sh, and then produces a log of events in svc.config.restore.prepare.

-execute
(Optional) Runs the command script svc.config.restore.sh, and produces a log of events in svc.config.restore.execute.log.

-fmt
(Optional) Specifies that the volume should be formatted before use. Includes the **-fmtdisk** option on all **mkvdisk** commands to be issued. You cannot specify **-fmt** with **-execute**.

-fmtdisk
(Optional) Specifies that the volume should be formatted before use. You cannot specify **-fmtdisk** with **-execute**.

-v on | off
Produces verbose output (on); the default is regular output (off).

-ver
(Required) Returns the version number for the **svcconfig** command.

Description

This command provides syntax help for svcconfig.

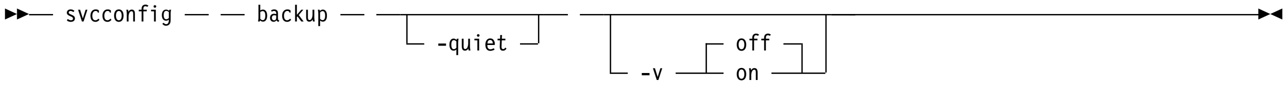
An invocation example

```
svcconfig -ver
svcconfig -?
svcconfig backup
```

backup

Use the **backup** command to back up the configuration. Enter this command any time after creating the clustered system (system).

Syntax



Parameters

-quiet
(Optional) Suppresses standard output (STDOUT) messages from the console.

-v on | off
(Optional) Displays normal (off, the default state) or verbose (on) command messages.

Description

The **backup** command extracts and stores configuration information from the system. The **backup** command produces the `svc.config.backup.xml`, `svc.config.backup.sh`, and `svc.config.backup.log` files, and saves them in the `/tmp` folder. The **.xml** file contains the extracted configuration information; the `.sh` file contains a script of the commands used to determine the configuration information; and the **.log** file contains details about command usage.

Note: If a previous `svc.config.backup.xml` file exists in the `/tmp` folder, it is archived as `svc.config.backup.bak`; only one archive file is stored in the `/tmp` folder.

The underscore character (_) prefix is reserved for backup and restore command usage; do not use the underscore character in any object names.

An invocation example

```
svcconfig backup
```

The resulting output:

```
No feedback
```

clear

Use the **clear** command to erase files in the `/tmp` directory that were previously produced by other **svcconfig** commands. You can enter this command any time after a clustered system (system) has been created.

Syntax

```
►►── svcconfig ── ── clear ── ┬──────────┬── ┬───────────┬── ┬────────────────┬──────────────────►◄
                              └─ -all ───┘   ├─ -q ──────┤   └─ -v ──┬─ on ──┬┘
                                             └─ -quiet ──┘           └─ off ─┘
```

Parameters

-all
Erases all configuration files.

-q | quiet
(Optional) Suppresses console output (STDOUT).

-v on | off
(Optional) Produces verbose output (`on`); the default is regular output (`off`).

Description

This command erases configuration files on the current configured node.

You can use the **clear** command without the **-all** parameter to erase files of the form:

```
/tmp/svc.config*.sh
/tmp/svc.config*.log
```

You can use the **clear** command with the **-all** parameter to erase files of the form:

```
/tmp/svc.config*.sh
/tmp/svc.config*.log
/tmp/svc.config*.xml
/tmp/svc.config*.bak
```

An invocation example

```
svcconfig clear -all
```

The resulting output:

```
No feedback
```

cron

Use the **cron** command to back up the configuration. Enter this command any time after creating the clustered system (system).

Syntax

```
►►── svcconfig ── ── cron ── ┬──────────┬─ ┬──────────────────┬──────────────────────►◄
                             └─ -quiet ─┘  │        ┌─ off ─┐ │
                                           └─ -v ───┴─ on ──┴─┘
```

Parameters

-q, -quiet
Suppresses standard output (STDOUT) messages from the console.

-v on, -v off
Displays normal (off, the default state) or verbose (on) command messages.

Description

This command generates configuration files and places them in the configuration files directory. The file svc.config.cron.xml_*(node)* contains configuration detail. The file svc.config.cron.log_*(node)* contains a log of events. The file svc.config.cron.sh_*(node)* contains a script of the commands used to determine the configuration.

Any pre-existing file svc.config.cron.xml_*(node)* is archived as svc.config.cron.bak_*(node)*. Only one such archive is kept.

The configuration files directory is /dumps.

An invocation example

```
svcconfig cron
svcconfig cron -q
svcconfig cron -v on
```

recover

Use the **recover** command to recover the clustered system configuration in two phases, the preparation phase and the execution phase. This is a component of T3 Recovery.

Syntax

```
►►── svcconfig ── ── recover ── ┬───────────┬─ ┬───────────┬─ ┬──────────────┬─ ──────────────►
                                ├─ -f ──────┤  ├─ -q ──────┤  ├─ -prepare ──┤
                                └─ -force ──┘  └─ -quiet ──┘  └─ -execute ──┘
```

```
►─┬──────────────────┬───────────────────────────────────────────────────────────►◄
  │        ┌─ off ─┐ │
  └─ -v ───┴─ on ──┴─┘
```

Parameters

-execute
(Optional) Runs the command script `svc.config.recover.sh` and produces a log of events in `svc.config.recover.execute.log`.

-f, -force
(Optional) Forces continued processing where possible.

-prepare
(Optional) Verifies the current configuration against the information in `svc.config.backup.xml` on the configuration to be recovered. Prepares commands for processing in `svc.config.recover.sh`, and produces a log of events in `svc.config.recover.prepare.log`.

-q, -quiet
(Optional) Suppresses console output (STDOUT).

-v on, -v off
(Optional) Produces verbose output (on); the default is regular output (off).

Description

The **recover** command recovers the target system configuration from the `svc.config.backup.xml` file, and associated `.key` files (if present) in the configuration files folder.

The recover operation is performed in two phases: prepare and execute. If neither the **-prepare** nor the **-execute** option is specified, the command performs both phases in sequence, producing only a single event log: `svc.config.recover.log`.

The configuration files directory is `/tmp`.

An invocation example

```
svcconfig recover -prepare
svcconfig recover -execute
```

restore

Use the **restore** command to restore the clustered system (system) to its previous configuration. This command uses the configuration files in the `/tmp` folder .

Syntax

```
►►── svcconfig ── ── restore ── ─┬──────────┬─ ─┬──────────┬─ ─┬──────────────────────────────┬──────►
                                 ├─ -f ─────┤   ├─ -q ─────┤   ├─ -prepare ──┬──────────────┬─┤
                                 └─ -force ─┘   └─ -quiet ─┘   │             ├─ -fmt ───────┤ │
                                                               │             └─ -fmtdisk ───┘ │
                                                               ├─ -execute ───────────────────┤
                                                               ├─ -fmt ───────────────────────┤
                                                               └─ -fmtdisk ───────────────────┘
```

```
►─ ──┬──────────────────┬──────────────────────────────────────────────►◄
     │       ┌─ off ─┐  │
     └─ -v ──┴─ on ──┴──┘
```

Parameters

-f | force
(Optional) Forces continued processing where possible.

-q | quiet
(Optional) Suppresses console output (STDOUT).

-prepare *-fmt* | *fmtdisk*
(Optional) Verifies the current configuration against the information in `svc.config.backup.xml`, prepares commands for processing in `svc.config.restore.sh`, and produces a log of events in `svc.config.restore.prepare`.

-execute
(Optional) Runs the command script `svc.config.restore.sh`, and produces a log of events in `svc.config.restore.execute.log`.

-fmt
(Optional) Specifies that the volume must be formatted before use. Includes the **-fmtdisk** option on all **mkvdisk** commands to be issued. You cannot specify **-fmt** with **-execute**.

-fmtdisk
(Optional) Specifies that the volume must be formatted before use. You cannot specify **-fmtdisk** with **-execute**.

-v on | off
(Optional) Produces verbose output (`on`); the default is regular output (`off`).

Description

The **restore** command restores the target system configuration from the `svc.config.backup.xml` file in the `/tmp` folder. If neither the **-prepare** nor the **-execute** option is specified, the command performs both phases in sequence, producing only a single event log: `svc.config.restore.log`.

The restore operation is also known as a T4 (Tier 4) Recovery, and can only be used on a system having just been started. The restore operation can not be used on a system having any nonautomatic objects configured, such as storage pools or volumes.

The restore operation is performed in two phases: prepare and execute.

The command pauses for eight minutes if any nodes are added during this process, informing the user of this at run-time.

An invocation example

```
svcconfig restore
```

The resulting output:

```
No feedback
```

An invocation example

```
svcconfig restore -prepare -fmt
```

The resulting output:

```
No feedback
```

An invocation example

```
svcconfig restore -execute
```

The resulting output:

```
No feedback
```

Chapter 7. Cloud commands

Use the cloud commands to create, change, or list details about cloud-related objects. Use the cloud commands to create, change, or list details about system cloud and the SAN Volume Controller system.

cfgcloudcallhome

Use the **cfgcloudcallhome** command to configure email and metering features by using an Internet Protocol (IP) quorum server as a Simple Mail Transfer Protocol (SMTP) server on a system. This command is for IBM Spectrum Virtualize for Public Cloud only.

Syntax

```
►►— cfgcloudcallhome — — -username — — -key — — -ip — — -ibmcustomer — — -ibmcountry ————————►◄
```

Parameters

-username
(Required) Specifies the IBM Cloud™ application programming interface (API) user name.

-key
(Required) Specifies the IBM Cloud API key.

-ip
(Required) Specifies the IP address of the IP quorum server.

-ibmcustomer
(Required) Specifies the customer number that is assigned when a software license is automatically added to the entitlement database. The value must be a 7 - 10-digit number.

-ibmcountry
(Required) Specifies the country ID used for entitlement and the call home system. The value is either a 3-digit number or blank.

Description

This command configures email and billing features by using an Internet Protocol (IP) quorum server as a Simple Mail Transfer Protocol (SMTP) server on a system.

An invocation example

```
# cfgcloudcallhome -username callhome1@de.ibm.com -key xxxxx -ip 192.168.0.1 -ibmcustomer 12345678 -ibmcountry 886
```

The following output is displayed:

```
None
```

cfgcloudstorage

Use the **cfgcloudstorage** command to configure IBM Cloud storage. This command is for IBM Spectrum Virtualize for Public Cloud only.

Syntax

```
►►— cfgcloudstorage — — -username — — -key — — -storage — — -srcportid ——————————————►◄
```

Parameters

-username
Specifies the IBM Cloud application programming interface (API) user name.

-key
Specifies the IBM Cloud API key.

-storage
Specifies the IBM Cloud storage name.

-srcportid
Specifies the node port ID.

Description

This command configures the IBM Cloud backend storage.

An invocation example

```
cfgcloudstorage
```

The following output is displayed:

```
No feedback
```

querycloudstoragecandidate

Use the **querycloudstoragecandidate** command to query the IBM Cloud storage candidate that is mapped to the system. This command is for IBM Spectrum Virtualize for Public Cloud only.

Syntax

```
►►— querycloudstoragecandidate — ┬—————————————┬— ┬————————┬—————————————————————————►◄
                                 └— -username —┘  └— -key —┘
```

Parameters

-username
(Optional) IBM Cloud API user name.

-key
(Optional) IBM Spectrum Virtualize for Public Cloud API key.

Description

This command queries the IBM Cloud storage candidate that is mapped to the clustered system.

The Table 20 table describes attribute values that can be displayed as output view data.

Table 20. **querycloudstoragecandidate** *output*

Attribute	Description
storage	Indicates the candidate storage name.
datacenter	Indicates the data center to which the storage belongs.

Table 20. **querycloudstoragecandidate** *output (continued)*

Attribute	Description
storage_type	Indicates either Endurance or Performance type. Storage type depends on the storage type when purchased.
iops	Indicates IOPS in total
capacity_gb	Indicates the capacity of storage when purchased.

An invocation example for querycloudstoragecandidate

```
$ querycloudstoragecandidate -usr qingyuanhou -key xxxxx
```

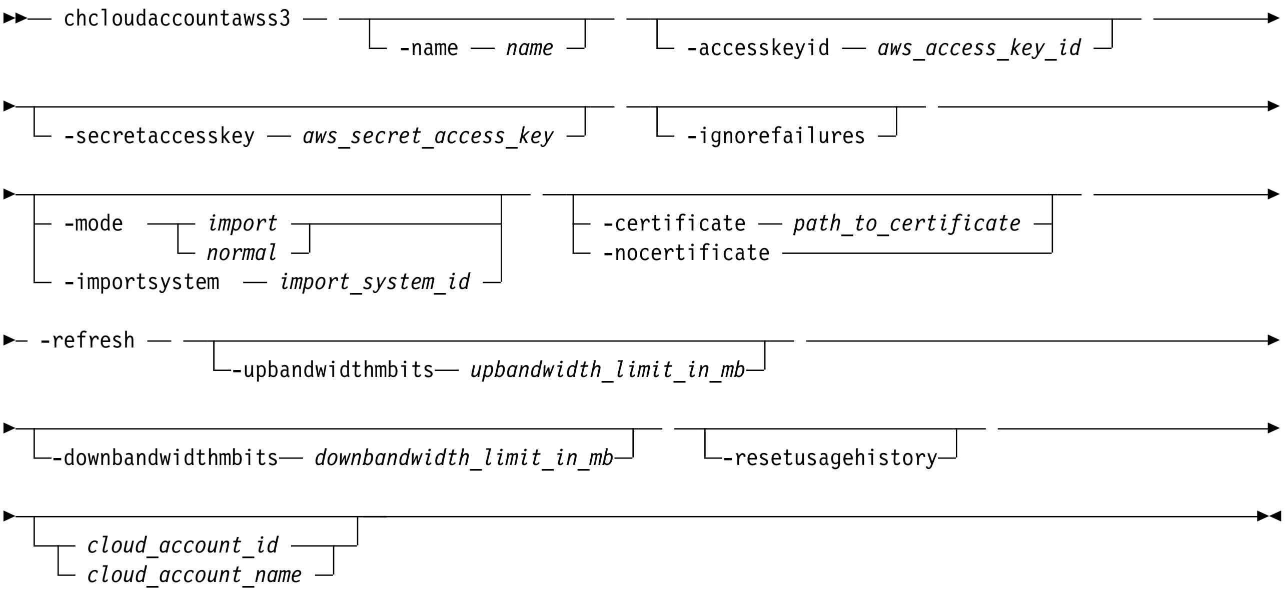
The following output is displayed:

```
storage                datacenter         storage_type       iops    capacity_gb

IBM01SEL571877-10      lon02              ENDURANCE          4000    100
IBM01SEL571877-11      lon02              ENDURANCE          4000    100
IBM01SEL571877-12      lon02              PERFORMANCE        40000   40
```

chcloudaccountawss3

Use the **chcloudaccountawss3** command to modify the cloud account (that uses Amazon S3 storage) parameters or mode.

Syntax


chcloudaccountawss3
-name name
-accesskeyid aws_access_key_id
-secretaccesskey aws_secret_access_key
-ignorefailures
-mode
import
normal
-importsystem import_system_id
-certificate path_to_certificate
-nocertificate
-refresh
-upbandwidthmbits upbandwidth_limit_in_mb
-downbandwidthmbits downbandwidth_limit_in_mb
-resetusagehistory
cloud_account_id
cloud_account_name

Parameters

-name *name*
(Optional) Specifies the new or modified cloud account name. The value must be an alphanumeric value.

-accesskeyid *aws_access_key_id*
(Optional) Specifies the value for the public part of the Amazon Web Services (AWS) access key. Use this access key to access cloud storage.

-secretaccesskey *aws_secret_access_key*
(Optional) Specifies the value for the private part of the Amazon Web Services (AWS) access key. This access key is for the AWS user that the system uses to access cloud storage.

-ignorefailures
(Optional) Changes the access key whether the new access key works.

-mode *import* | *normal*
(Optional) Specifies the new or modified cloud account mode. The values are import or normal.

-importsystem *import_system_id*
(Optional) Specifies that the system's data be imported.

Note: You must specify -mode *import* first.

-certificate *path_to_certificate*
(Optional) Specifies the path for the SSL certificate to use when you authenticate to the new or modified cloud account storage. The value must be an alphanumeric string of 1 - 255 characters (in base64-encoded PEM format).

-nocertificate
(Optional) Specifies that the custom SSL certificate that was used to authenticate to the new or modified cloud account storage is used to stop the system.

-refresh
(Optional) Specifies a refresh of the system import candidates. If the account is in import mode, this parameter specifies a refresh of the data available for import.

-downbandwidthmbits *downbandwidth_limit_in_mb*
(Optional) Specifies the download bandwidth limit in megabits per second (Mbps). The value must be a number 1 - 10240.

-upbandwidthmbits *upbandwidth_limit_in_mb*
(Optional) Specifies the upload bandwidth limit in megabits per second (Mbps). The value must be a number 1 - 10240.

-resetusagehistory
(Optional) Resets the usage history (to 0). Storage consumption that reflects the space that is consumed on the cloud account is cumulative, which means that it remains in the current day row (the 0th row).

cloud_account_id | *cloud_account_name*
(Required) Specifies the cloud account ID or name to modify. The value for the ID must be a number and the value for the name must be an alphanumeric string.

Description

This command modifies the parameters for the cloud account (created by using **mklcloudaccountawss3**) that uses Amazon S3 storage.

The **-mode** parameter, the **-refresh** parameter, and any of the user credentials parameters groups are mutually exclusive.

This command fails and no changes are made if the supplied credentials do not provide authentication. Credentials include:

- **-accesskeyid**
- **-secretaccesskey**
- **-certificate** or **nocertificate**

For example, if the network is down then the system cannot confirm that a new secret access key is valid, the command fails. Specify **-ignorefailures** to overwrite this feature. If you specify invalid credentials but you do specify **-ignorefailures**, an online account becomes offline and an error is generated in the error log that describes the authentication failure.

The secret access key is sensitive system information and is stored in encrypted form. It is not available in the system dumps and in the audit log it is replaced with six hash ("#") symbols.

If you specify this command against an offline account and these new details enable the account to start working (for example, you enter an expired password) the account becomes online.

You must change the mode if the account is not being used by any system volumes. A mode change requires the account to be online and the system be able to communicate with the cloud server.

Note: You can have a maximum of:

- One cloud account per clustered system (system)
- 1024 volumes with cloud snapshots enabled
- 256 cloud snapshots per volume
- 512 volume groups

An invocation example

```
chcloudaccountawss3 -name myamazon cloudaccount0
```

The resulting output:

```
No feedback
```

An invocation example

```
chcloudaccountawss3 -mode import -importsystem 000002007D40A162 0
```

The resulting output:

```
No feedback
```

An invocation example

```
chcloudaccountawss3 -upbandwidthmbits 100 -downbandwidthmbits 100 cloudaccount0
```

The resulting output:

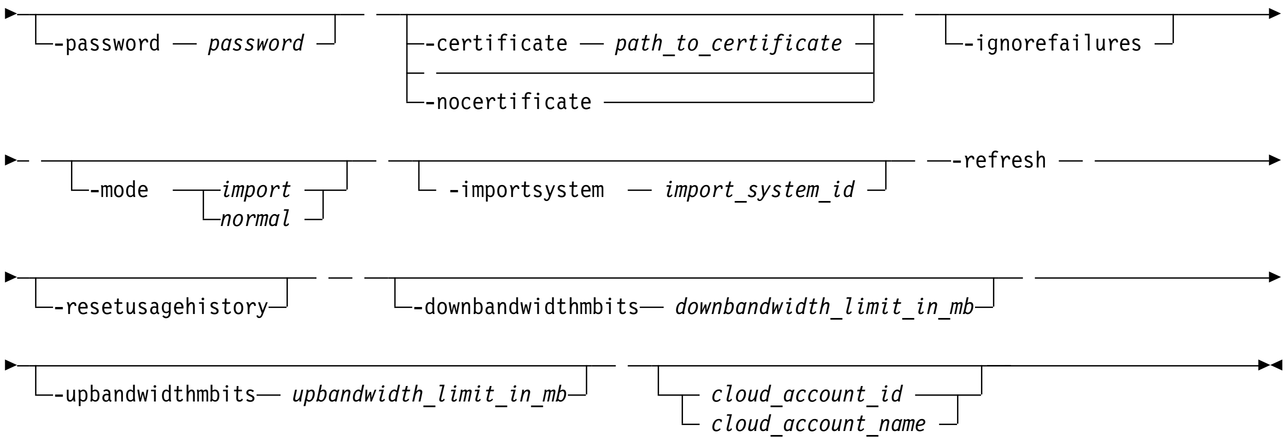
```
No feedback
```

chcloudaccountswift

Use the **chcloudaccountswift** command to modify the cloud account (that uses OpenStack Swift storage) parameters or mode.

Syntax

```
►►── chcloudaccountswift ── ┬───────────────┬── ┬───────────────────────┬── ──────────────────►
                            └─-name ── name ─┘   └─-keystone ──┬─ yes ─┬─┘
                                                               └─ no ──┘

►─┬─────────────────────────────────┬── ┬─────────────────────────┬── ────────────────────────►
  └─-endpoint ── https_endpoint_URL ─┘   └─-username ── user_name ─┘
```


-password — password
-certificate — path_to_certificate
-nocertificate
-ignorefailures
-mode
import
normal
-importsystem — import_system_id
-refresh
-resetusagehistory
-downbandwidthmbits— downbandwidth_limit_in_mb
-upbandwidthmbits— upbandwidth_limit_in_mb
cloud_account_id
cloud_account_name

Parameters

-name *name*
(Optional) Specifies the new or modified OpenStack name you must use to access cloud account storage. The value must be an alphanumeric value.

-keystone *yes* | *no*
(Optional) Specifies that keystone authentication is used. The values are *yes* or *no*.

-endpoint *https_endpoint_URL*
(Optional) Specifies the URL (that the system uses to access object storage) to change for the cloud account. If OpenStack Keystone authentication is used, the URL specified must be the URL for the Keystone authentication. If Keystone authentication is not used, the URL specified must be the URL for the Swift account. The value must be 8 - 128 characters and must be a valid URL address.

-username *user_name*
(Optional) Specifies the OpenStack user name that the system must use to access cloud account storage.

-password *password*
(Optional) Specifies the password value to use to authenticate to cloud storage. For IBM Cloud accounts, this password is the application programming interface (API) key. The value must be 1 - 64 alphanumeric characters and it must not begin or end with a space.

-certificate *path_to_certificate*
(Optional) Specifies the path for the SSL certificate to use when you authenticate to new or modified cloud account storage. The value must be an alphanumeric string of 1 - 255 characters (in base64-encoded PEM format).

-nocertificate
(Optional) Specifies that the custom SSL certificate that was used to authenticate to the new or modified cloud account storage is used to stop the system.

-ignorefailures
(Optional) Specifies that the access key is changed whether or not the new access key works.

-mode *import* | *normal*
(Optional) Specifies the new or modified cloud account mode. The values are `import` or `normal`.

-importsystem *import_system_id*
(Optional) Specifies that the system's data be imported.

Note: You must specify -mode *import* first.

-refresh
(Optional) Specifies a refresh of the system import candidates. If the account is in `import` mode, this parameter specifies a refresh of the data available for import.

-downbandwidthmbits *downbandwidth_limit_in_mb*
(Optional) Specifies the download bandwidth limit in megabits per second (Mbps). The value must be a number 1 - 10240.

-upbandwidthmbits *upbandwidth_limit_in_mb*
(Optional) Specifies the upload bandwidth limit in megabits per second (Mbps). The value must be a number 1 - 10240.

-resetusagehistory
(Optional) Resets the usage history (to `0`). Storage consumption that reflects the space that is consumed on the cloud account is cumulative, which means that it remains in the current day row (the `0th` row).

cloud_account_id | *cloud_account_name*
(Required) Specifies the cloud account ID or name to modify. The value for the ID must be a number and the value for the name must be an alphanumeric string.

Description

This command modifies the parameters for the cloud account (created by using **mklcloudaccountswift**) that uses OpenStack Swift storage.

At least one parameter must be set.

The **-mode** parameter, the **-refresh** parameter, and any of the user credentials parameters groups are mutually exclusive. Credentials include:

- **-keystone**
- **-endpoint**
- **-username**
- **-password**
- **-certificate** or **nocertificate**

The command fails if the supplied authentication credentials are unsuccessful. For example, if the network is down the system cannot confirm that the `secretaccesskey` is valid (and the command fails). Specify **-ignorefailures** to override this feature. If you specify incorrect credentials and the **-ignorefailures** parameter, an online account becomes offline and an error is generated in the log describing the authentication failure.

The password is treated as sensitive system information. It is stored in an encrypted form and not available in system dumps. In the audit log, it is replaced with six hash ("#") symbols.

If a certificate is supplied and the command succeeds, the certificate file is deleted from the local file system.

If you specify this command against an offline account and these new details enable the account to start working (for example, you enter a new password against an expired password) the account becomes online.

You can change the mode if the account is not being used by any system volumes. A mode change requires the account to be online and the system be able to communicate with the cloud server.

Note: You can have a maximum of:

- One cloud account per clustered system (system)
- 1024 volumes with cloud snapshots enabled
- 256 cloud snapshots per volume
- 512 volume groups

An invocation example

```
chcloudaccountswift -certificate /tmp/new-cert.pem -ignorefailures myswift
```

The resulting output:

```
No feedback
```

An invocation example

```
chcloudaccountswift -mode import -importsystem 000002007D40A162 0
```

The resulting output:

```
No feedback
```

An invocation example

```
chcloudaccountawss3 -username newuser -password simpsons 0
```

The resulting output:

```
No feedback
```

An invocation example

```
chcloudaccountswift -upbandwidthmbits 100 -downbandwidthmbits 100 cloudaccount0
```

The resulting output:

```
No feedback
```

lscloudaccount

Use the **lscloudaccount** command to display information about the configured cloud accounts.

Syntax

```
►►─ lscloudaccount ─ ┬──────────┬─ ┬─────────────────────┬─ ┬──────────────────────┬──►◄
                     └─ -nohdr ─┘  └─ -delim ─ delimiter ─┘  └─┬─ cloud_account_id ───┤
                                                               └─ cloud_account_name ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character.

If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

`cloud_account_id` | *`cloud_account_name`*
(Optional) Specifies the name or ID for the detailed view of of the account.

Description

This command displays information about configured cloud accounts.

This table provides the attribute values that can be displayed as output view data.

Table 21. lscloudaccount output

Attribute	Description
id	Indicates the cloud account ID. The value is a number.
name	Indicates the cloud account name. The value is an alphanumeric string.
type	Indicates the cloud account provider. The values are `awss3` or `swift`.
status	Indicates the cloud account status. The values are `online` or `offline`.
mode	Indicates the cloud account mode. The values are `normal` or `import`.
active_volume_count	Indicates the number of volumes in the system that use the account. The value must be a number.
backup_volume_count	Indicates the number of volumes that are backed up to the cloud account. The value must be a number.
import_system_id	Indicates the system ID for the system from where the data is being imported. The value must be a 16-character uppercase hexadecimal number (or blank).
import_system_name	Indicates the system name from where the data is being imported. The value must be an alphanumeric string (or blank).
error_sequence_number	Indicates an error (for offline accounts). The value must be a number (or blank).
refreshing	Indicates whether the system is refreshing its cloud storage view (for `import` mode accounts). The values are `yes` or `no`.
backup_timestamp	Indicates the timestamp for the most recent backup. The value must be in the format `YYMMDDHHMMSS` (or blank).
certificate	Indicates whether SSL is configured for an account that uses certificates. The values are `yes` or `no`.
certificate_expiry	Indicates the time and date that a certificate expires. The value must be blank or be in this format: `Dec 7 10:07:59 2015 GMT`
endpoint	Indicates the endpoint URL for `swift` accounts. The value must be a valid URL (or blank).
awss3_bucket_prefix	Indicates the bucket prefix that is being used for S3 accounts. The value must be a valid bucket prefix (or blank).
awss3_access_key_id	Indicates the user access key ID for S3 accounts. The value must be a valid access key ID (or blank).
awss3_region	Indicates the region that is chosen for cloud storage for S3 accounts. The value must be for a valid AWS region (or blank).
swift_keystone	Indicates whether keystone authentication is in use. The value must be `yes` or `no`.
swift_container_prefix	Indicates the container prefix for Swift accounts. The value must be a valid container prefix or blank.
swift_tenant_name	Indicates the tenant name that is used for authentication for `swift` accounts. The value must be a valid tenant name (or blank).

Table 21. lscloudaccount output (continued)

Attribute	Description
swift_user_name	Indicates the user name that is used for authentication for `swift` accounts. The value must be a valid user name (or blank).
encrypt	Indicates the encryption status for the cloud account. The values are `yes` and `no`.

A concise invocation example

```
lscloudaccount
```

The resulting output:

```
id name      type  status mode   active_volume_count backup_volume_count import_system_id import_system_name error_sequence
0  importer  swift online import 2                   123                 000002007D40A162 cluster1
```

A detailed invocation example

```
lscloudaccount 1
```

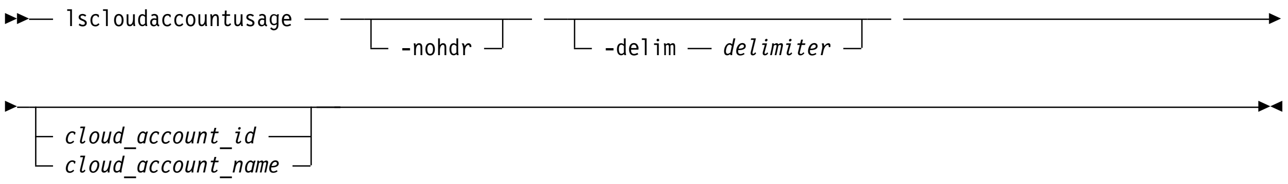
The resulting output:

```
id 0
name vardyja
type swift
status online
mode normal
active_volume_count 0
backup_volume_count 1
import_system_id
import_system_name
error_sequence_number
refreshing no
backup_timestamp 151021114002
certificate yes
certificate_expiry Dec  7 10:07:59 2017 GMT
endpoint https://thesecurecloud.company.com:4000/auth/v3.0
awss3_bucket_prefix
awss3_access_key_id
awss3_region
swift_keystone yes
swift_container_prefix svc-1
swift_tenant_name mytenant
swift_user_name storeman
```

lscloudaccountusage

Use the **`lscloudaccountusage`** command to list usage information about configured cloud storage accounts.

Syntax



Parameters

-nohdr

(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*

(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

cloud_account_id | *cloud_account_name*

(Optional) Specifies the cloud account to list details for. The ID value must be a number and the name value must be an alphanumeric string.

Description

This command displays usage information about configured cloud storage accounts. The information involves chargeable resource usage.

This table provides the attribute values that can be displayed as output view data.

Table 22. lscloudaccountusage output

Attribute	**Description**
id	Indicates the ID for the cloud account. The value must be a number 0 - 4294967295.
name	Indicates the name for the cloud account. The value must be an alphanumeric string.
date	Indicates the date for the system data that is displayed. Each row shows usage of one day. The value must be in `YYYYMMDD` format. This value is computed relative to the current configured system date. The date in the very first entry must equate to the current date. If you manually change the system date, the changes are not reflected in the output for the date field until midnight (the time `00:00`). If you change your system date got to accommodate time zone changes, it is reflected in the output instantly. After midnight, any subsequent entry is for the next 24-hour period.
upload_data_mb	Indicates the uploaded data for one day. The value must be a number 0 - 18446744073709551615.
download_data_mb	Indicates the downloaded data for one day. The value must be a number 0 - 18446744073709551615.
storage_consumed_gb	Indicates the volume of data that is stored in this cloud account. The value must be a number 0 - 18446744073709551615.

Note: For a detailed view, there are 180 rows. Each row has information corresponding to one full day, and every field reflects activity for that day except for the `storage_consumed_gb`, which is cumulative. The most recent entry reflects the current day.

An invocation example

```
lscloudaccountusage
```

The resulting output:

```
id    name             date          upload_data_mb  download_data_mb     storage_consumed_gb
0     cloudaccount0    20151023        194560        900                  6700
1     cloudaccount1    20151023        204800        1500                 10700
```

An invocation example

```
lscloudaccountusage 0
```

The resulting output:

```
id                  name                       date             upload_data_mb  download_data_mb    storage_consumed_gb
0                   cloudaccount0              20151023         194560          900                 6687
0                   cloudaccount0              20151022         3584000         150                 6495
0                   cloudaccount0              20151021         1024            17152               3010
```

lscloudaccountimportcandidate

Use the **lscloudaccountimportcandidate** command to list information about systems that have data that is stored in the cloud accounts that are defined on this system.

Syntax

```
►►── lscloudaccountimportcandidate ── ┬─────────┬─ ┬──────────────────────┬─ ─────────────────►◄
                                      └─ -nohdr ┘  └─ -delim ── delimiter ┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim : on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

Description

This command lists information about systems that have data that is stored in the cloud accounts that are defined on this system.

This command provides information about the valid options for chcloudaccount -import. To refresh the view by reloading what is on the cloud server, specify chcloudaccount -refresh.

This table provides the attribute values that can be displayed as output view data.

Table 23. lscloudaccountimportcandidate output

Attribute	Description
cloud_account_id	Indicates the ID for the cloud account that contains data from another system. The value must be a numeric string.
cloud_account_name	Indicates the name for the cloud account that contains data from another system. The value must be an alphanumeric string.
import_system_id	Indicates the system ID of the system that has data on the cloud account. The value must be a 16-character string in hexadecimal uppercase.
import_system_name	Indicates the system name of the system that has data on the cloud account. The value must be an alphanumeric string.
backup_volume_count	Indicates the number of volumes that are backed up by the imported system. The value must be a numeric string.
backup_size	Indicates the approximate amount of cloud storage that is in use by snapshots from the imported system.
backup_timestamp	Indicates the timestamp of the most recent volume backup (by the other system). The value must be in the `YYMMDDHHMMSS` format or be blank. This value is displayed in UNIX time.

An invocation example

```
lscloudaccountimportcandidate
```

The resulting output:

```
cloud_account_id cloud_account_name import_system_id import_system_name backup_volume_count backup_size backup_timestamp
0                my_amazon          00002007D40A162  cluster1           0                   0.00GB
0                my_amazon          00002007F42E813  cluster2           44                  15.25TB     151008084203
```

mkcloudaccountawss3

Use the **mkcloudaccountawss3** command to configure a new cloud account that uses Amazon S3 object storage.

Syntax

```
►►── mkcloudaccountawss3 ── ┬─────────────────┬ ── -bucketprefix ── bucket_prefix ── ──────────►
                            └─ -name ── name ─┘

►─ -accesskeyid ── aws_access_key_id ── ── -secretaccesskey ── aws_secret_access_key ── ──────►

►─┬───────────────────────────────────────┬─ ─┬────────────────────────────────────────┬─ ────►
  └─ -certificate ── path_to_certificate ─┘   └─-upbandwidthmbits── upbandwidth_limit_in_mb─┘

►─┬──────────────────────────────────────────────────┬─ ─┬─────────────────────────┬─ ───────►
  └─-downbandwidthmbits── downbandwidth_limit_in_mb─┘   └─-region── aws_region─┘

►─┬─────────────────────────┬───────────────────────────────────────────────────────────────►◄
  └─ -encrypt ──┬─ yes ──┬──┘
                └─ no ───┘
```

Parameters

-name *name*
(Optional) Specifies the name for the cloud account. The value must be an alphanumeric string.

-bucketprefix *bucket_prefix*
(Required) Specifies the prefix for the S3 bucket names that the system uses. The value must be a lower-case alphabetic string 3 - 58 characters long (with no dot or period at the end of the string, and no dot or period next to another dot or period).

-accesskeyid *aws_access_key_id*
(Required) Specifies the public part of the Amazon Web Services (AWS) access key credential of the AWS user that the system use to access the cloud storage. The value must be a 20-character alphanumeric string of uppercase letters and numbers.

-secretaccesskey *aws_secret_access_key*
(Required) Specifies the non-public part of the AWS access key credential that the system use to access the cloud storage. The value must be a 40-character alphanumeric string (That can contain slashes, or "/").

-certificate *path_to_certificate*
(Optional) Specifies the path to an SSL certificate authority (CA) certificate for AWS S3. The value must be an alphanumeric string of 1 - 255 characters (in base64-encoded PEM format).

-upbandwidthmbits *upbandwidth_limit_in_mb*
(Optional) Specifies the upload bandwidth limit in megabits per second (Mbps). The value must be a number 1 - 10240.

-downbandwidthmbits *downbandwidth_limit_in_mb*
(Optional) Specifies the download bandwidth limit in megabits per second (Mbps). The value must be a number 1 - 10240.

-region *aws_region*
(Optional) Specifies the AWS region to use to access the cloud account and store data.

-encrypt *yes* | *no*
(Optional) Specifies whether to encrypt the data in the cloud account. By default, encryption is enabled unless you specify -encrypt no.

Description

This command configures a new cloud account that uses Amazon S3 object storage.

An invocation example

```
mkcloudaccountawss3 -name myamazon
                    -bucketprefix svc_backups
                    -accesskeyid AKIAIOSFODNN7EXAMPLE
                    -secretaccesskey wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
                    -upbandwidthmbits 100
                    -downbandwidthmbits 100
```

The resulting output:

```
Cloud Account, id [0], successfully created
```

Note: If the system contains an encrypted cloud account that uses USB encryption, a USB flash drive with the system master key must be present in the configuration node before the cloud account can move to the online state. This requirement is necessary when the system is powered down, and then restarted.

mkcloudaccountswift

Use the **mkcloudaccountswift** command to configure a new cloud account that uses OpenStack Swift object storage.

Syntax

```
►►── mkcloudaccountswift ── ─┬──────────────┬─ ─┬────────────┬─ ─────────────────────────────►
                             └─-name── name ─┘   └─-keystone──┘

►──-containerprefix── container_prefix ── ──-endpoint──┬─ http_endpoint_URL ──┬── ───────────────►
                                                       └─ https_endpoint_URL ─┘

►──-tentantname── tenant_name ── ──-username── user_name ── ──-password── password ── ──────────►

►─┬──────────────────────────────────────┬─┬───────────────────────────────────────────────┬─ ──►
  └─-certificate── path_to_certificate ──┘ └─-upbandwidthmbits── upbandwidth_limit_in_mb──┘

►─┬───────────────────────────────────────────────────┬─┬───────────────────────┬─────────────►◄
  └─-downbandwidthmbits── downbandwidth_limit_in_mb──┘ └─ -encrypt ──┬─ yes ──┬─┘
                                                                     └─ no ───┘
```

Parameters

-name *name*
(Optional) Specifies the account identifier. The value must be an alphanumeric string.

-keystone
(Optional) Specifies that the system authenticate with OpenStack Keystone. If you do not specify this parameter, the system authenticates with OpenStack TempAuth.

-containerprefix *container_prefix*
(Required) Specifies the Swift container names the system uses or creates. The value must be 1 - 12 characters and contain no spaces or slashes.

-endpoint *http_endpoint_URL* | *https_endpoint_URL*
(Required) Specifies the URL that the system uses to access object storage.

If Keystone authentication is used, it is the URL of the Keystone service, probably ending with `v2.0`. Otherwise, it is the URL of the Swift service.

-tentantname *tenant_name*
(Required) Specifies the OpenStack tenant the system uses to access cloud storage. The value must be a 1 - 64 alphanumeric characters that contain no spaces.

-username *user_name*
(Required) Specifies the OpenStack user name the system uses to access cloud storage. The value must be 1 - 255 alphanumeric characters with no spaces.

-password *password*
(Required) Specifies the password that the system uses to access cloud storage. For IBM Cloud accounts, this password is the application programming interface (API) key. The value must be 1 - 64 alphanumeric characters and it must not begin or end with a space.

-certificate *path_to_certificate*
(Optional) Specifies the file path for the object storage server SSL certificate. The value must be:

- 1 - 255 alphanumeric characters with no period or dot next to another period or dot, and no period or dot at the start or end of the specified value
- In base64-encoded PEM format

-upbandwidthmbits *upbandwidth_limit_in_mb*
(Optional) Specifies the upload bandwidth limit in megabits per second (Mbps). The value must be a number 1 - 10240.

-downbandwidthmbits *downbandwidth_limit_in_mb*
(Optional) Specifies the download bandwidth limit in megabits per second (Mbps). The value must be a number 1 - 10240.

-encrypt *yes* | *no*
(Optional) Specifies whether to encrypt the data in the cloud account. By default, encryption is enabled unless you specify `-encrypt no`.

Description

This command configures a new cloud account that uses OpenStack Swift object storage.

An invocation example

```
mkcloudaccountswift -containerprefix svc_backups
                    -endpoint https://lon02.objectstorage.cloud.net/auth/v1.0
                    -tenantname mytenant
                    -username jamivard
                    -password WKF84FAQRKLOICDF53LANBWKF84FAQRKLOICDF53LANBEXAMPLEEXAMPLEEXAMPL
                    -upbandwidthmbits 100
                    -downbandwidthmbits 100
```

The resulting output:

```
Cloud Account, id [0], successfully created
```

Note: If the system contains an encrypted cloud account that uses USB encryption, a USB flash drive with the system master key must be present in the configuration node before the cloud account can move to the online state. This requirement is necessary when the system is powered down, and then restarted.

rmcloudaccount

Use the **rmcloudaccount** command to delete a cloud account from the system.

Syntax

```
►►— rmcloudaccount — ┬ cloud_account_id ──┬ ─────────────────────────────────────►◄
                     └ cloud_account_name ┘
```

Parameters

cloud_account_id | *cloud_account_name*
(Required) Specifies the cloud account to remove. The ID value must be a number and the name value must be an alphanumeric string.

Description

This command deletes a cloud account from the system. If no systems have volume data that is stored in the account, the containers are deleted from the cloud storage.

As long as there are no volumes on this system that are using the cloud account, the command deletes the account. If there are no volumes that are left in the account, the system attempts to delete its containers. If you cannot connect to the cloud server, the containers are not deleted. If the command times out, the deletion proceeds asynchronously and the account object is removed.

An invocation example

```
rmcloudaccount VardyAmazAcct
```

The resulting output:

```
No feedback
```

testcloudaccount

Use the **testcloudaccount** command to run diagnostics against the cloud account and report status on the results.

Syntax

```
►►── testcloudaccount ── ─┬─ cloud_account_id ───┬──────────────────────────────────────►◄
                          └─ cloud_account_name ─┘
```

Parameters

cloud_account_id | *cloud_account_name*
(Required) Specifies the cloud account to test. The ID value must be a number and the name value must be an alphanumeric string.

Description

This command runs diagnostics against the cloud account and reports status, which includes network connectivity, authentication, and cloud storage usage.

This command can be run against an online or offline account.

- If the command is run successfully against an offline account, the account becomes online.
- If the command is run unsuccessfully against an online account, the account becomes offline.

An invocation example

```
testcloudaccount MyVardyAccount
```

The resulting output:

```
Cloud Account, id [0], successfully tested
```

Chapter 8. Clustered system commands

Use the clustered system (system) commands to monitor and modify systems and their properties.

addnode (SAN Volume Controller only)

Use the **addnode** command to add a new (candidate) node to an existing clustered system (system). Enter this command any time after a system is created. If you are adding a node to a system, make sure that the model type of the new node is supported by the system code (code) version of the existing system. If the model type is not supported by the code, upgrade the system to a code version that supports the model type of the new node.

Syntax

```
►►— addnode — ┬ -panelname — — panel_name ┬ ┬─────────────────────────┬ ──────────────►
              └ -wwnodename — — wwnn_name ┘  └ -name — — node_name ┘

►┬ -spare — spare_name ───────────┬ ┬─────────────────────────┬──────────────────────►◄
 └ -iogrp — ┬ iogroup_name ┬──────┘  └ -site — ┬ site_name ┬─┘
            └ iogroup_id ──┘                   └ site_id ──┘
```

Parameters

-panelname *panel_name*
: (Required if you do not specify the **-wwnodename** parameter) Specifies the node that you want to add to a system by the name that is displayed in the management GUI, the service assistant, or displayed by specifying **lsnodecandidate**. You cannot use this parameter with the **-wwnodename** parameter.

  Note: If `panel_name` is not supplied, it applies to the node on which the command is running.

-wwnodename *wwnn_name*
: (Required if you do not specify the **-panelname** parameter) Specifies the node that you want to add to the system by the worldwide node name (WWNN). You cannot use this parameter with the **-panelname** parameter.

-name *node_name*
: (Optional) Specifies a name for the node that you want to add to the system. You can use this name in subsequent commands to refer to the node, instead of using the node ID.

  Note: Node names that are supplied with the **-name** parameter on the **addnode** and **chnode** commands must not already be in use as node names or as node `failover_names`.
  If you assign a name, this name is displayed as the node name from then on. If you do not assign a name, a default name is used. The default name that is used depends on whether the node is replacing one that was previously deleted. When a node is deleted, its name is retained in the I/O group as the failover name of its partner node. If no nodes remain in an I/O group, no failover names are retained. Only one failover name can be stored for each node. If you add a node into an I/O group that has a retained failover name and do not specify a node name, the retained failover name is assigned to this node. If you do not specify a name and there is no retained failover name, the name that is assigned has the format `node`*X*.

  Important: The iSCSI Qualified Name (IQN) for each node is generated using the system and node names. If you are using the iSCSI protocol and the target name for this node is already active on its partner node, and iSCSI hosts are attached to it. Adding the node with a different name changes the IQN of this node in the system and might require reconfiguration of all iSCSI-attached hosts.

-spare *spare_name*
: (Optional) Specifies that the node that is being added is a spare node and not an I/O node group member. You cannot specify this parameter with **-iogrp**.

-iogrp *iogroup_name* | *iogroup_id*
: (Required) Specifies the I/O group to which you want to add this node. You cannot specify this parameter with **-spare**.

-site *site_name* | *site_id*
: (Optional) Specifies the numeric site value or site name of the new node.

If the system topology is hyperswap and the I/O group has a configured node, this new node must be located within the same site. If no configured nodes exist in the I/O group (but volumes are defined in the I/O group that are in active-active relationships) this new node must be located within the same site as any node that was previously in that I/O group.

Remember:

- This parameter must be specified whether the system topology is set to stretched or hyperswap.
- If the system topology is stretched and the I/O group has a configured node, this new node must be in another site location.

Description

Note: The **addnode** command is a SAN Volume Controller command. For Storwize V7000, use the **addcontrolenclosure** command.

This command adds a node to the system. You can obtain a list of candidate nodes (nodes that are not already assigned to a system) by typing lsnodecandidate. You cannot add a node with less memory than any potential partner nodes that are in the I/O group.

Note: The **lsnodecandidate** command is a SAN Volume Controller command. For Storwize V7000, use the **lscontrolenclosurecandidate** command.

You can create thin-provisioned volumes in a data reduction storage pool on all node types. Compressed volumes in a data reduction storage pool must be created in an I/O group with node types that support compression. Nodes that support compression can be added to an I/O group that contains compressed volumes.

You cannot use this command if the new node is:

- Not capable of encryption but the existing I/O group partner is capable of encryption.
- Not capable of encryption but storage pools exist with encryption keys that include MDisks that are not self-encrypting.
- Capable of encryption but the node has no encryption license.

Note: This command is successful only if the node-enclosure system ID matches the system, or is blank.

When the first thin or compressed volume in a data reduction pool is created for an IO group, the IO group sets CPU parameters based on the lowest number of CPU resources available based on the nodes in the I/O group. A new node with less CPU resources cannot be added to the I/O group.

Before you add a node to the system, you must check to see if any of the following conditions are true. If the following conditions exist, failure to follow the procedures that are documented here might result in the corruption of all data that is managed by the system.

- Is the new node being used to replace a failed node in the system?
- Does the node being added to the system use physical node hardware that has been used as a node in another system, and are both system recognized by the same hosts?

If any of the previous conditions are true, you must take the following actions:

1. Add the node to the same I/O group that it was previously in. You can use the command-line interface command **lsnode** or the management GUI to determine the WWNN of the system nodes.
2. Shut down all of the hosts that use the system before you add the node back into the system.
3. Add the node back to the system before the hosts are restarted. If the I/O group information is unavailable or it is inconvenient to shut down and restart all of the hosts that use the system, you can do the following:
   a. On all of the hosts that are connected to the system, unconfigure the Fibre Channel adapter device driver, the disk device driver, and the multipathing driver before you add the node to the system.
   b. Add the node to the system, and then reconfigure the Fibre Channel adapter device driver, the disk device driver, and multipathing driver.

If you are adding a node to a system, take the following actions:

1. Ensure that the model type of the new node is supported by the code level for the system. If the model type is not supported by the system code, you must upgrade the system to a version of code that supports the model type of the new node.
2. Record the node serial number, the WWNN, all WWPNs, and the I/O group to which the node is added. You might need to use this information later. Having it available can prevent possible data corruption if the node must be removed from and re-added to the clustered system.

Note: Transparent cloud tiering can be enabled on a system if every node on the system supports it. If a system supports transparent cloud tiering, you cannot add nodes that do not support it to the system.

Other considerations when you add a node to a system:

When you add a node to the system by using the **addnode** command or the system GUI, you must confirm whether the node has previously been a member of the system. If it has, follow one of these two procedures:

- Add the node to the same I/O group that it was previously in. You can determine the WWNN of the nodes in the system by using the **lsnode** command.
- If you cannot determine the WWNN of the nodes in the cluster, call the support team to add the node back into the system without corrupting the data.

When a node is added to a system, it displays a state of adding. It can take 30 minutes for the node to be added to the system, particularly if the version of code that is associated with the node changed.

Attention: If the node remains in the adding state for more than 30 minutes, contact your support representative to assist you in resolving this issue.

When a node is deleted, its name is retained in an I/O group as the failover name of its partner node. If no nodes remain in an I/O group, no failover names are retained.

The **addnode** command fails if you specify a name that is either an existing node name or a retained failover name, or if the system has a configuration that exceeds the limits for the node that is being added. Specify a different name for the node that is being added.

Compressed or thin deduplicated volumes can be added only to systems in which all nodes support deduplicated volumes. You can only add nodes that support deduplicated volumes to a system that contains compressed or thin deduplicated volumes. Nodes can only be added to a system that contains compressed or thin deduplicated volumes if that new node can support the amount of memory that is allocated for data deduplication in the target I/O group.

An invocation example

```
addnode -wwnodename 5005076801e08b -iogrp io_grp0
```

The resulting output:

```
Node, id [6], successfully added
```

An invocation example

```
addnode -panelname 123456 -iogrp 1 -site 2
```

The resulting output:

```
Node, id [6], successfully added
```

An invocation example

```
addnode -wwnodename 5005076801e08b -iogrp io_grp0 -site site1
```

The resulting output:

```
Node, id [6], successfully added
```

An invocation example

```
addnode -panelname 123456 -spare
```

The resulting output:

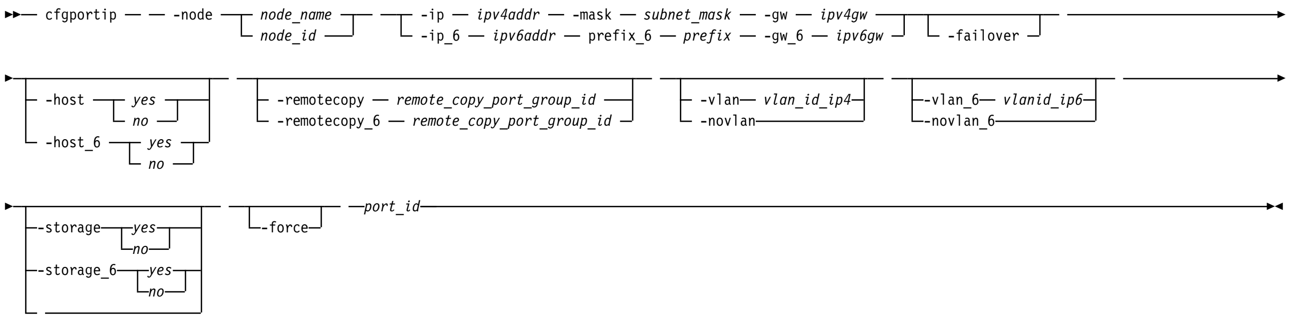
```
Node, id [7], successfully added
```

cfgportip

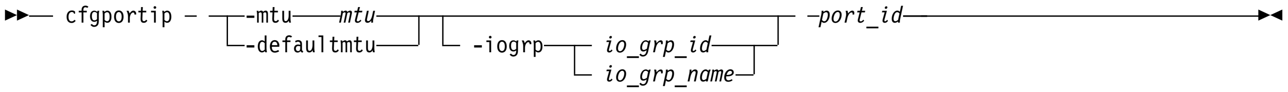
Use the **cfgportip** command to assign an Internet Protocol (IP) address to each node Ethernet port for Internet Small Computer System Interface (iSCSI) input/output (I/O).

Syntax

For Internet Protocol Version 4 (IPv4) and Internet Protocol Version 6 (IPv6):



For maximum transmission unit (MTU):



Parameters

-node *node_name* | *node_id*
(Required) Specifies which node has the Ethernet port that the IP address is being assigned to.

Note: This parameter is required for setting a port IP address. It cannot be used with the **-mtu** parameter.

-ip *ipv4addr*
(Required) Sets the Internet Protocol Version 4 (IPv4) address for the Ethernet port. You cannot use this parameter with the **ip_6** parameter.

-ip_6 *ipv6addr*
(Required) Sets the Internet Protocol Version 6 (IPv6) address for the Ethernet port. You cannot use this parameter with the **ip** parameter.

-gw *ipv4addr*
(Required) Sets the IPv4 gateway IP address. You cannot use this parameter with the **gw_6** parameter.

-gw_6 *ipv6gw*
(Required) Sets the IPv6 default gateway address for the port. You cannot use this parameter with the **gw** parameter.

-mask *subnet_mask*
(Required) Sets the IPv4 subnet mask. You cannot use this parameter with the **prefix_6** parameter.

-prefix_6 *prefix*
(Required) Sets the IPv6 prefix. You cannot use this parameter with the **mask** parameter.

-failover
(Optional) Specifies that the IP address belongs to the partner node in the I/O group. If the partner node is not configured or offline, the address is configured and presented by this node. When another node comes online in the I/O group, the failover address is presented by that node.

If the partner node is online, do not use this option.

-mtu *mtu* | **-defaultmtu**
(Required) Specifies the maximum transmission unit (MTU). The default is 1500, with a maximum of 9000. An MTU of 9000 allows you to save CPU utilization for packets of 4 KB and over in size. The increased MTU provides you with improved Internet Small Computer System Interface (iSCSI) performance. Specify **-defaultmtu** to use the default value.

Notes: This parameter has the following restrictions:

- This parameter must be used when you are setting or changing the clustered system (system) MTU value.
- This parameter cannot be used with the **-node** parameter.

-iogrp *iogrp*
(Optional) Specifies the I/O group that contains the nodes to modify.

-host *yes* | *no*
(Optional) Specifies the IPv4 address that is used for host attach (the existing system settings are retained). Specifying:

- *yes* reports the IPv4 address to hosts during target discovery (default)
- *no* turns off this report (IPv4 addresses are not reported during host discovery).

-remotecopy *remote_copy_port_group_id*
(Optional) Specifies the IPv4 address that is used for the remote copy function. Remote copy includes HyperSwap, Metro Mirror, and Global Mirror. It also specifies the ID for the associated port group. These IDs are numerical values (`0`, `1`, or `2`) that specify that IP addresses on a system can be part of a

partnership for a login. To form a login, IP addresses must be in the same port group. The default is `0`, which indicates that the port is not available for partnerships.

Important: To add or delete ports to or from a replication group, make sure that the partnership that uses that port group is in a stopped state.

`-host_6` *`yes`* **|** *`no`*
(Optional) Specifies the IPv6 address that is used for host attach (the existing system settings are retained). Specifying:

- *yes* reports the IPv6 address to hosts during target discovery (default).
- *no* turns off this report (IPv6 addresses are not reported during host discovery).

Note: Turning off host attach settings for an IP address that is set to `yes` is disruptive because all host iSCSI sessions to that IP address are logged out.

`-remotecopy_6` *`remote_copy_port_group_id`*
(Optional) Specifies the IPv6 address that is used for the remote copy function. Remote copy includes HyperSwap, Metro Mirror, and Global Mirror. It also specifies the ID for the associated port group. These IDs are numerical values (`0`, `1`, or `2`) that specify that IP addresses on a system can be part of a partnership for a login. To form a login, IP addresses must be in the same port group. The default is `0`, which indicates that the port is not available for partnerships.

Important: To add or delete ports to or from a replication group, make sure that the partnership that uses that port group is in a stopped state.

`-vlan` *`vlanid_ip4`*
(Optional) Sets the virtual local area network (VLAN) ID for a IPv4 address that is configured for iSCSI host attach or remote copy function. Remote copy includes HyperSwap, Metro Mirror, and Global Mirror. The VLAN ID for an IPv4 type address can be specified only if the IP address for that port is set. VLAN tagging is disabled for any IP address, so a VLAN ID must be specified by using **`-vlan`** to turn on VLAN tagging.

Remember: Use **`-vlan`** with caution. You can:

- Reset VLAN settings, which can disrupt port communication (connection) with hosts or systems (including resetting the VLAN ID for an active iSCSI or IP partnership)
- Reset a VLAN value for a port that does not have VLAN tagging or does not have a configured IP address

The VLAN ID can be set for the failover port that uses the **`-failover`** attribute.

`-novlan`
(Optional) Disables VLAN tagging for an IPv4 address for an Ethernet port (which means no VLAN tag is associated with that port).

`-vlan_6` *`vlanid_ip6`*
(Optional) Sets the virtual local area network (VLAN) ID for a IPv6 address that is configured for iSCSI host attach or remote copy function. Remote copy includes HyperSwap, Metro Mirror, and Global Mirror. The VLAN ID for an IPv6 type address can be specified only if the IP address for that port is set. VLAN tagging is disabled for any IP address, so a VLAN ID must be specified by using **`-vlan`** to turn on VLAN tagging.

Remember: Use **`-vlan_6`** with caution:

- Resetting VLAN settings can disrupt port communication (connection) with hosts or systems, including resetting the VLAN ID for an active iSCSI or IP partnership.
- You can reset a VLAN tag for a port that does not have VLAN tagging or does not have a configured IP address.

The VLAN ID can be set for the failover port by using the **`-failover`** attribute.

-novlan_6
(Optional) Disables Virtual local area network (VLAN) tagging for an IPv6 address for an Ethernet port (which means no VLAN tag is associated with that port).

Remember: Use **-novlan_6** with caution:

- Resetting VLAN settings can disrupt port communication (connection) with hosts or systems, including resetting the VLAN ID for an active iSCSI or IP partnership.
- You can reset a VLAN tag for a port that does not have VLAN tagging or does not have a configured IP address.

-storage *yes* | *no*
(Optional) Specifies whether an IPv4 address can be used for the backend storage attach function. The value *yes* indicates that this IPv4 address can be used for iSCSI target discovery and backend storage connectivity. You must specify *no* (default) if you are not using the storage attach IP address. If the IPv4 address associated with a specific port (on a node) is changed, the existing storage attachment settings are retained. The values are *yes* and *no*.

-storage_6 *yes* | *no*
(Optional) Specifies whether an IPv6 address can be used for the backend storage attach function. The value *yes* (the default) indicates that this IPv6 address can be used for iSCSI target discovery and backend storage connectivity. You must specify *no* if you are not using a storage attachment IP address. If the IPv6 address associated with a specific port (on a node) is changed, the existing storage attachment settings are retained. The values are *yes* and *no*.

-force
(Optional) Forces an IP address change for a node Ethernet port even if this causes iSCSI backend controllers being removed or MDisks going into a degraded or offline state.

Important: Use the force attribute rarely to prevent a loss of access to a node or a MDisk.

Changing IP address attributes can cause MDisks to go into a degraded state for some time. If a source IP address is in use for iSCSI backend controller connectivity, changing the IP address or the subnet mask or the IP gateway removes existing sessions and establishes new sessions. During this phase, the MDisks visible though the source port that is reconfigured go to degraded state for a short while until new sessions are established.

You can use the -force flag to go ahead with the reconfiguration if you understand all of the risks involved. If you are unsure of what might happen, use the force attribute only under the direction of your support personnel.

If you are adding a new I/O group in the system, you might see message CMMVC8915E. When you configure iSCSI IP addresses on a new I/O group, make sure that you assign the IP address to an unconfigured port and use the -force flag for IP assignment.

port_id
(Required) Specifies which port (1, 2, 3, or 4) to apply changes to.

Description

The **cfgportip** command either sets the IP address of an Ethernet port for iSCSI, or configures the MTU of a group of ports. This command assigns either an IPv4 or IPv6 address to a specified Ethernet port of a node. The IP address is used for iSCSI I/O. Use the **chsystemip** command to assign clustered system IP addresses.

Remember: When IP addresses are configured with the same remote replication port group ID (for redundancy) to each node of an I/O group, make sure that the same Ethernet port for both nodes is used during configuration. MTU is set by using symmetric Ethernet ports from the same I/O group. To make sure alternative remote replication port groups work with the same MTU settings, symmetric Ethernet ports must be configured for remote replication port groups.

For an IPv4 address, the **ip**, **mask**, and **gw** parameters are required. All of the IPv4 IP parameters must be specified to assign an IPv4 address to an Ethernet port.

For an IPv6 address, the **ip_6**, **prefix_6**, and **gw_6** parameters are required. All of the IPv6 IP parameters must be specified to assign an IPv6 address to an Ethernet port.

If an IP address is specified for a host, the specified port can be discovered by hosts using the iSNS server (or other discovery mechanisms such as SendTargets). These IP addresses are not reported to partner systems in order to create TCP sessions that are used for remote copy. These ports also cannot be used for login to and SendTargets based discovery of backend iSCSI Storage controllers.

IP addresses that are specified for remote copy cannot be discovered by hosts, which means they cannot be used for host attachment. These ports are not reported to partner systems in order to create TCP sessions for remote copy. These ports also cannot be used to log in to and for SendTargets when considering discovery of backend iSCSI Storage controllers.

After IP configuration, *host_port_group_id* is automatically assigned to the iSCSI ports. Host port grouping groups the ports that have the same speed and ensures that no more than four ports are discovered by a host. Additional *host_port_group_id* criteria include:

- A *host_port_group_id*is an automatic grouping of ports that is designated by an integer. Host port group IDs are unique across I/O groups.
- Each host port group ID contains a maximum of four ports.
- All ports within a host port group ID have identical speeds.
- Identical host port group IDs are assigned to the failover port. If a *host_port_group_id* is already assigned to a failover port, the same *host_port_group_id* are assigned to a local port.
- Enabling -host flag to *yes* assigns the *host_port_group_id*. If on a port with host flag *no* , host flag is set to *yes* , resulting in assignment of a *host_port_group_id* to a port.
- Disabling the flag to *no* removes the host port group id associated with a iSCSI port.

IP addresses that are specified for storage cannot be discovered by hosts, which means they cannot be used for host attachment. These IP addresses are not reported to partner systems to create and set up TCP sessions for remote copy.

To use the same IP address for both host I/O and backend storage attach functions (but not for remote copy):

- The **-host** parameter must be set to *yes*.
- The **-storage** parameter must be set to *yes*.
- The **-remotecopy** parameter must be set to *no*.

In these instances, these IP addresses can be discovered by hosts. These IP addresses can also be used for backend storage controller discovery and login for iSCSI based migration and virtualization.

To use the same IP address for both backend storage attach functions and remote copy functions (but not for host I/O operations):

- The **-storage** parameter must be set to *yes*.
- The **-remotecopy** parameter must be specified with the required remote copy port group ID.
- The **-host** parameter must be set to *no*.

In such cases, these IP addresses can be used to discover and connect to backend iSCSI storage controllers. These IP addresses can also be used for IP-based remote copy.

To use the same IP address for both host I/O and remote copy functions (but not for backend storage attach functions):

- The **-host** parameter must be set to *yes*.
- The **-remotecopy** parameter must be invoked with the required remote copy port group ID.
- The **-storage** parameter must be set to *no*.

In such cases, these IP addresses can be discovered by hosts as well as used for IP-based Remote Copy but not for backend storage attach.

Use the **lsportip** command with the optional **ethernet_port_id** parameter to list the port IP addresses for the specified port.

Remember:

If **cfgportip** is used to modify the IP address that is associated with a specific Ethernet port without specifying a new VLAN ID, the new (modified) IP address inherits the existing VLAN ID setting of the earlier IP address (IPv4 or IPv6).

An invocation example for IPv4

```
cfgportip -node 1  -ip 9.8.7.1 -gw 9.0.0.1 -mask 255.255.255.0 1
```

The resulting output:

```
No feedback
```

An invocation example for IPv6

```
cfgportip -node 1  -ip_6 3:3:0:4::0  -gw_6 ffe8::0  -prefix_6 64 2
```

The resulting output:

```
No feedback
```

An invocation example to set an MTU of 1600 on port 1 in I/O group 0

```
cfgportip  –mtu 1600 -iogrp 0 1
```

The resulting output:

```
No feedback
```

An invocation example to set the MTU to its default value

```
cfgportip  –defaultmtu -iogrp 0 1
```

The resulting output:

```
No feedback
```

An invocation example configuring a new IPv4 address for IP-based replication

```
cfgportip -node 1 -ip 9.8.7.1 -gw 9.0.0.1 -mask 255.255.255.0 –remotecopy 1 –host no -host_6 no 1
```

The resulting output:

```
No feedback
```

An invocation example configuring a new IPv4 address for host attach

```
cfgportip -node 1 -ip 9.8.7.1 -gw 9.0.0.1 -mask 255.255.255.0 –host yes 1
```

The resulting output:

```
No feedback
```

An invocation example configuring replication for an existing IPv6 address

```
cfgportip -node 1 –remotecopy_6 2 1
```

The resulting output:

```
No feedback
```

An invocation example configuring host attach for a new IPv6 address

```
cfgportip -node 1 –ip_6 2001:db8::1:0:0:1 –host_6 yes 1
```

The resulting output:

```
No feedback
```

An invocation example configuring a new IPv4 address with the VLAN ID 105

```
cfgportip -node 1 -ip 9.8.7.1 -gw 9.0.0.1 -mask 255.255.255.0 –vlan 105 1
```

The resulting output:

```
No feedback
```

An invocation example for configuring a new IPv6 address with the VLAN ID 1063

```
cfgportip -node 1 -ip_6 2001:db8::1:0:0:101 -prefix_6 64 -gw_6 2001:db8::1:0:0:1 -vlan_6 1063 1
```

The resulting output:

```
No feedback
```

An invocation example for configuring a new IPv4 address for the backend storage attach function using iSCSI

```
cfgportip -node 1 -ip 9.8.7.1 -gw 9.0.0.1 -mask 255.255.255.0 –storage yes -remotecopy 0 –host no 1
cfgportip -node 1 -ip 9.8.7.1 -gw 9.0.0.1 -mask 255.255.255.0 –storage yes –host no 1
```

The resulting output:

```
No feedback
```

An invocation example for configuring a new IPv4 address for host attach only

```
cfgportip -node 1 -ip 9.8.7.1 -gw 9.0.0.1 -mask 255.255.255.0 –host yes -storage no 1
cfgportip -node 1 -ip 9.8.7.1 -gw 9.0.0.1 -mask 255.255.255.0 –host yes -storage no -remotecopy 0 1
cfgportip -node 1 -ip 9.8.7.1 -gw 9.0.0.1 -mask 255.255.255.0 1
```

The resulting output:

```
No feedback
```

An invocation example for configuring a new IPv4 address for IP-based replication

```
cfgportip -node 1 -ip 9.8.7.1 -gw 9.0.0.1 -mask 255.255.255.0 -storage no –remotecopy 1 –host no 1
```

The resulting output:

```
No feedback
```

An invocation example for configuring the storage attach function for a new IPv6 address

```
cfgportip -node 1 –ip_6 2001:db8::1:0:0:1 –storage_6 yes 1
cfgportip -node 1 –ip_6 2001:db8::1:0:0:1 1
```

The resulting output:

```
No feedback
```

An invocation example for changing the storage specification for an existing IPv6 address

```
cfgportip -node 1 –storage_6 no 1
cfgportip -node 1 –storage_6 yes 1
```

The resulting output:

```
No feedback
```

chbanner

Use the **chbanner** command to configure the login message that is displayed during CLI Secure Shell (SSH) login.

Syntax

```
►►─ chbanner ─ ┬──────────────────────┬─ ┬───────────┬─ ┬────────────┬─ ┬──────────┬─ ─────►◄
               └─ -file ─file_path─┘   └─ -enable ─┘   └─ -disable ─┘   └─ -clear ─┘
```

Parameters

-file *file_path*
(Optional) Specifies the path to the file on the configuration node that contains the new login message.

-enable
(Optional) Enables the login message.

-disable
(Optional) Disables the login message.

-clear
(Optional) Clears the login message.

Description

This command configures the login message that is displayed during CLI SSH login. Use this command for warnings or disclaimers or anything else you need to display in your login screen before logging in.

The file that contains the login message must be copied to the configuration node before you specify `chbanner -file`. If a configuration node failover occurs between copying the file that contains the login message and running the command, the temporary file must be copied to the new configuration node.

To set a login message that uses a SAN administrator's workstation:

1. Use a suitable text editor to create the message and save the file with a recognizable name.
2. Use a secure copy client to copy the file to the configuration node of the system to be configured.
3. Specify the management IP address of the system that is to be configured.
4. Log into the system to be configured.
5. Use the **chbanner** command to set the login message.

An invocation example

```
 chbanner -file /tmp/loginmessage
```

The detailed resulting output:

```
No feedback
```

An invocation example

```
chbanner -enable
```

The detailed resulting output:

```
No feedback
```

An invocation example

```
chbanner -disable
```

The detailed resulting output:

```
No feedback
```

An invocation example

```
chbanner -clear
```

The detailed resulting output:

```
No feedback
```

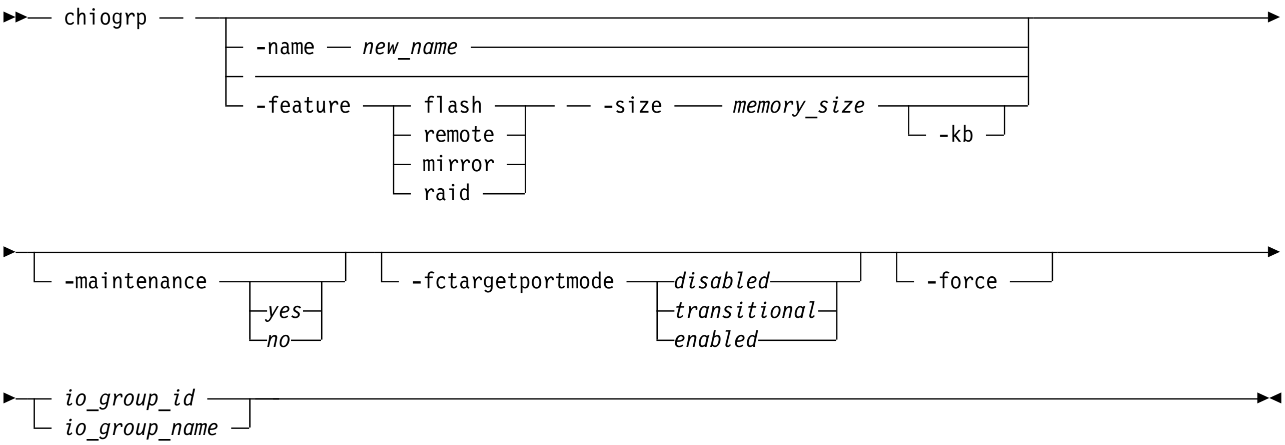
chcluster (Discontinued)

Attention: The **chcluster** command has been discontinued. Use the **chsystem** command instead.

chiogrp

Use the **chiogrp** command to modify the name of an I/O group, or the amount of memory that is available for RAID arrays, Copy Services, FlashCopy services, or volume mirroring operations.

Syntax



Parameters

-name *new_name*
(Optional) Specifies the name to assign to the I/O group. The **-name** parameter cannot be specified with the **-feature**, **-size**, or **-kb** parameters.

-feature flash | remote | mirror | raid
(Optional) Specifies the feature to modify the amount of memory for RAID arrays, Copy Services, or volume mirroring. You must specify this parameter with the **-size** parameter. You cannot specify this parameter with the **-name** parameter.

- `flash` specifies the amount of memory that is used for FlashCopy.

- `remote` specifies the amount of memory that is used for remote copy processing. Remote copy includes Metro Mirror, Global Mirror, and HyperSwap.
- `mirror` specifies the amount of memory that is used for volume mirroring operations.
- `raid` specifies the amount of memory that is used for RAID arrays.

Note: Specifying **`remote`** changes the amount of memory that is available for remote copy processing. Any volume that is in a remote copy relationship uses memory in its I/O group, including master and auxiliary volumes, and volumes that are in inter-system or intra-system relationships.

`-size` *`memory_size`*
(Optional) Specifies the amount of memory that is available for the specified RAID arrays, Copy Services, or volume mirroring function. Valid input is `0` or any integer. The default unit of measurement for this parameter is megabytes (MB); you can use the kilobytes **`-kb`** parameter to override the default. You must specify this parameter with the **`-feature`** parameter. You cannot specify this parameter with the **`-name`** parameter.

`-kb`
(Optional) Changes the units for the **-size** parameter from megabytes (MB) to kilobytes (KB). If you specify this parameter, the **`-size`** *memory_size* value must be any number divisible by 4. You must specify this parameter with the **`-feature`** and **`-size`** parameters. You cannot specify this parameter with the **`-name`** parameter.

`-maintenance` *`yes`* **|** *`no`*
(Optional) Specifies whether the I/O group must be in maintenance mode. The I/O group must be placed in maintenance mode while performing service procedures on storage enclosures. After you enter maintenance mode, it continues until either:

- It is explicitly cleared.
- Thirty minutes elapse.

Note: Changing the maintenance mode on any I/O group changes the maintenance mode on all I/O groups.

`-fctargetportmode` *`disabled`* **|** *`transitional`* **|** *`enabled`*
(Optional) Specifies the Fibre Channel (FC) host port mode of the I/O group. The values are *disabled*, *transitional*, or *enabled*. The *transitional* state is an intermittent state where both the virtual ports and physical ports are enabled.

`-force`
(Optional) Specifies that an FC host port be disabled or enabled, even if disruption to host I/O might occur as a result. You can only specify **`-force`** with **`-fctargetportmode`**.

Important: Specifying **`-force`** might result in a loss of access. Use it only under the direction of your product support information.

`io_group_id` **|** *`io_group_name`*
(Required) Specifies the I/O group to modify. You can modify an I/O group by using the **`-name`** or the **`-feature`** parameter.

Description

The **`chiogrp`** command modifies the name of an I/O group or the amount of memory that is available for RAID arrays, Copy Services, or volume mirroring.

Use the **`-feature`** and **`-size`** parameters (together) to change the amount of memory available in the I/O group to one of the following types:

- FlashCopy
- Volume mirroring

- RAID
- Remote copy, including Metro Mirror, Global Mirror, and HyperSwap.

For example:

```
chiogrp -feature flash -size 40 0
```

You can assign a name to an I/O group or change the name of a specified I/O group. You can change the amount of memory that is available for RAID arrays, Copy Services, or volume mirroring operations by specifying the **-feature flash | remote | mirror** parameter - and a memory size. For volume mirroring and Copy Services (Flash Copy®, Metro Mirror, Global Mirror, and HyperSwap), memory is traded against memory that is available to the cache.

The amount of memory can be decreased or increased. Consider the following memory sizes when you use this command:

- The default amount of memory for FlashCopy is 20 MB.
- The default amount of memory for remote copy (which includes Metro Mirror, Global Mirror, and HyperSwap) is 20 MB.
- The default memory size for mirrored volumes is 20 MB.
- The default memory size for RAID arrays is 40 MB.
- The maximum amount of memory that can be specified for FlashCopy is 512 MB. For 64-bit systems, the maximum is 2048 MB.
- The maximum amount of memory for remote copy (which includes Metro Mirror, Global Mirror, and HyperSwap) is 512 MB.
- The maximum memory size that can be specified for mirrored volumes is 512 MB.
- The maximum memory size for RAID arrays is 512 MB.

The maximum combined amount of memory across all features is 552 MB.

Note: For 64-bit systems, the maximum is 2600 MB. Some systems that are running 64-bit mode might have 2 GB of bitmap space to use for FlashCopy, which is enough for 4 PB of data space to be used per I/O group. For example, Metro Mirror, Global Mirror, Volume Mirroring, and RAID share 552 MB of bitmap space, which is enough to use 1080 PB of data space per I/O group. Older systems, such as those running 32-bit code, might be subject to a 740 MB limit.

Table 24 demonstrates the amount of memory that is required for RAID arrays, Copy Services, and volume mirroring. Each 1 MB of memory provides the following volume capacities and grain sizes:

Table 24. Memory required for RAID arrays, Copy Services, and volume mirroring

Feature	Grain size	1 MB of memory provides the following volume capacity for the specified I/O group
Metro Mirror and Global Mirror	256 KB	2 TB of total Metro Mirror and Global Mirror volume capacity
HyperSwap	256 KB	2 TB of total HyperSwap volume capacity **Note:** For 2 TB of HyperSwap volume capacity, 1 MB must be assigned in each caching I/O group.
FlashCopy	256 KB	2 TB of total FlashCopy source volume capacity
FlashCopy	64 KB	512 GB of total FlashCopy source volume capacity

Table 24. Memory required for RAID arrays, Copy Services, and volume mirroring (continued)

Feature	Grain size	1 MB of memory provides the following volume capacity for the specified I/O group
Incremental FlashCopy	256 KB	1 TB of total Incremental FlashCopysource volume capacity
Incremental FlashCopy	64 KB	256 GB of total Incremental FlashCopy source volume capacity
Volume mirroring	256 KB	2 TB of mirrored volumes

Table 25 provides an example of RAID level comparisons with their bitmap memory cost, where *MS* is the size of the member drives and *MC* is the number of member drives.

Table 25. RAID level comparisons

Level	Member count	Approximate capacity	Redundancy	Approximate bitmap memory cost
RAID-0	1-8	MC * MS	None	(1 MB per 2 TB of MS) * MC
RAID-1	2	MS	1	(1 MB per 2 TB of MS) * (MC/2)
RAID-5	3-16	(MC-1) * MS	1	1 MB per 2 TB of MS with a strip size of 256 KB; double with strip size of 128 KB.
RAID-6	5-16	less than (MC-2 * MS)	2	
RAID-10	2-16 (evens)	MC/2 * MS	1	(1 MB per 2 TB of MS) * (MC/2)
Note: There is a margin of error on the approximate bitmap memory cost of approximately 15%. For example, the cost for a 256 KB strip size for RAID-5 is ~1.15 MB for the first 2 TB of MS.				

For multiple FlashCopy targets, you must consider the number of mappings. For example, for a mapping with a 256 KB grain size, 8 KB of memory allows one mapping between a 16 GB source volume and a 16 GB target volume. Alternatively, for a mapping with a 256 KB grain size, 8 KB of memory allows two mappings between one 8 GB source volume and two 8 GB target volumes.

After you create a FlashCopy mapping, if you specify an I/O group other than the I/O group of the source volume, the memory accounting goes towards the specified I/O group, not towards the I/O group of the source volume.

Scenario 1

If the I/O group contains:

- At least one 8 GB node.
- At least one thin-provisioned or compressed volume in a data reduction pool.
- And you try to set the FlashCopy bitmap size for that I/O group to at least 1.5 GB.

The command fails due to insufficient resources available.

An invocation example to create a new I/O group `testiogrpone`

```
chiogrp -name testiogrpone io_grp0
```

The resulting output:

```
No feedback
```

An invocation example for changing the amount of Flash Copy® memory in `io_grp0` to `30` MB

```
chiogrp -feature flash -size 30 io_grp0
```

The resulting output:

```
No feedback
```

An invocation example for changing the amount of RAID memory in I/O group `0` to `512` MB

```
chiogrp -feature raid -size 512 0
```

The resulting output:

```
No feedback
```

`chiscsiiqn` (SAN Volume Controller only)

Use the **`chiscsiiqn`** command to change the Internet Small Computer Systems Interface (iSCSI) qualified name (or IQN).

Syntax

```
►►── chiscsiiqn ── ┬───────────────────────┬─ ─┬─ node_name ─┬──────────────────────────►◄
                   ├─ -iqn ── iscsi_iqn ─┤    └─ node_id ───┘
                   └─ -noiqn ──────────────┘
```

Parameters

-iqn *iscsi_iqn*
: (Optional) Specifies the iSCSI IQN to change. The value must be an alphanumeric string with no more than 79 characters.

-noiqn
: (Optional) Specifies deletion of the iSCSI initiator name that is stored in the node.

node_name | *node_id*
: (Required) Specifies the node name or ID to change. The node name is an alphanumeric string, and the node ID is a number.

Description

This command changes the iSCSI IQN. You must synchronize the iSCSI IQN and the node vital product data (VPD) (that is used as the iSCSI initiator name) to connect backend iSCSI storage. You can specify either **`-iqn`** or **`-noiqn`**, but you cannot specify both simultaneously.

An invocation example

```
svctask chiscsiiqn  -iqn iqn.2009-05.ibmcloud.com:test.node3  3
```

The following output is displayed:

```
No feedback
```

An invocation example

```
svctask chiscsiiqn  -noiqn  3
```

The following output is displayed:

```
No feedback
```

chiscsistorageport

Use the **chiscsistorageport** command to change authentication parameters, such as setting authentication credentials, removing authentication parameters, or updating credentials.

Syntax

```
>>-- chiscsistorageport -- -+----------+- -+-----------+- -+-----------------------+- ------------>
                            '- -force -'   '- -noauth -'   '- -node --+- id ---+-'
                                                                      '- name -'

>-+-----------------------------------+- -+--------------------------------+- ---------------------->
  '- -username -- target_user_name -'     '- -chapsecret -- target_chap -'

>- lsiscsistorageport-rowid ---------------------------------------------------------------------><
```

Parameters

-force
(Optional - Spectrum Virtualize for Public Cloud only) When used, specify **-force** only with the **-noauth** parameter to force the clearance of authentication credentials for all initiator nodes in a single command.

-noauth
(Optional) Clears all authentication parameters for a session. For IBM Spectrum Virtualize for Public Cloud only, specify **-node** with **-noauth** to clear the credentials per initiator node. If **-node** is not specified, **-noauth** requires **-force** to clear the credentials for all initiator nodes.

Note: The **-noauth** parameter cannot be used with other parameters. IBM Spectrum Virtualize for Public Cloud is an exception. You must specify **-noauth** with **-force** to clear authentication for all initiator nodes, or with **-node** to clear authentication per initiator node, but not with other parameters.

-node *id* | *name*
(Optional - Spectrum Virtualize for Public Cloud only) Specifies the ID or name of a node in the system. The value must be an alphanumeric string.

-username *target_user_name*
(Optional) Specifies the target controller user name. The value must be an alphanumeric string up to 256 characters.For IBM Spectrum Virtualize for Public Cloud only, when **-node** is specified, the value of **-username** must be an alphanumeric string up to 32 characters. Otherwise, the value of **-username** must be an alphanumeric string up to 256 characters.

If the target controller requires a *target_user_name* and *target_chap* for discovery, the user name for the target controller must be specified.

Note: Changing the **-username** for a target controller can be a disruptive operation, so exercise caution when you change the authentication details for the session. Be sure to make the controller side authentication credentials changes before you change the authentication credentials for the session.

Some controllers might require that you use the iSCSI qualified name (IQN) user name for discovery. The IQN of each node is picked up automatically and used if required.

-chapsecret *target_chap*
(Optional) Specifies the Challenge-Handshake Authentication Protocol (CHAP) secret *target_chap* required for discovery of the target iSCSI controller. The value must be an alphanumeric string up to 80 characters. For IBM Spectrum Virtualize for Public Cloud only, when you specify **-node**, the value

of **-chapsecret** must be an alphanumeric string up to 32 characters. Otherwise, the value of **-chapsecret** must be an alphanumeric string up to 80 characters. This keyword is required when not using the **-noauth** parameter.

Note: Changing the **-chapsecret** for a target controller can be a disruptive operation, so exercise caution when you change the authentication details for the session. Be sure to make the controller side authentication credentials changes before you change the authentication credentials for the session.

lsiscsistorageport-rowid
(Required) Specifies the row ID of an existing **lsiscsistorageport** output row.

Description

The **chiscsistorageport** command operates on a row ID specified by the output of **lsiscsistorageport** command. Because some storage controllers do not drop the existing active session after you change the authentication credentials, this command forces the session to drop and reconnect to confirm that the changed authentication credentials work.

Note: You cannot change the session mode between target-specific authentication and initiator node-specific authentication by using this command.

An invocation example to clear authentication for an iSCSI session

This example shows how to clear the authentication details of a session. For example, first specify **lsiscsistorageport** to list iSCSI port information:

```
lsiscsistorageport
id   port_id target_ipv4 target_ipv6 target_iscsiname controller_id controller_name iogroup_list status site_id site_name
5     2      10.10.10.1              IQN1             1             ctlr1           1:1:1:1      full
```

The following example shows how to clear authentication where the target has a single user name or CHAP secret. This example applies to all products except for the IBM Spectrum Virtualize for Public Cloud product.

```
chiscsistorageport -noauth 5
```

The result is clearing the authentication of iSCSI sessions from all initiator nodes.

For IBM Spectrum Virtualize for Public Cloud only, the following example uses the **-force** parameter to show how to clear authentication where the target has a user name or CHAP secret per initiator node.

```
chiscsistorageport -force -noauth 5
```

The result is clearing the authentication of iSCSI sessions from all initiator nodes.

For IBM Spectrum Virtualize for Public Cloud only, to clear credentials for a specific node, specify **-node** with **-noauth**. To clear credentials for all nodes, specify **-force** instead of **-node** with **-noauth**.

An invocation example to change an existing user name

The following example shows how to change an existing user name where the target has a single user name or CHAP secret. The **-username** parameter requires the **-chapsecret** parameter. This example applies to all products except for the IBM Spectrum Virtualize for Public Cloud product.

```
chiscsistorageport -username superman -chapsecret abcd 5
```

The result is changing the authentication of iSCSI sessions from all initiator nodes.

For IBM Spectrum Virtualize for Public Cloud only, the following example shows how to change an existing user name where the target has a user name or CHAP secret per initiator node (node1 in this example).

```
chiscsistorageport -username superman -chapsecret batman -node node1 5
```

The result is changing the authentication of iSCSI sessions from initiator node node1.

An invocation example to change an existing CHAP secret

The following example shows how to change an existing **-chapsecret** where the target has a single user name or CHAP secret. This example applies to all products except the IBM Spectrum Virtualize for Public Cloud product.

```
chiscsistorageport -chapsecret batman 5
```

The result is changing the authentication of iSCSI sessions from all initiator nodes.

For IBM Spectrum Virtualize for Public Cloud only, specify the node name to change an existing chapsecret where the target has a user name or CHAP secret per initiator node (node1 in this example).

```
chiscsistorageport  -chapsecret batman -node node1 5
```

The result is changing the authentication of iSCSI sessions from initiator node node1.

An invocation example to change an existing user name and CHAP secret

This example shows how to change an existing **-username** and **-chapsecret** where the target has a single user name or CHAP secret. This example applies to all products except the IBM Spectrum Virtualize for Public Cloud product.

```
chiscsistorageport -username superman -chapsecret batman 5
```

The result is changing authentication of iSCSI sessions from all initiator nodes.

For IBM Spectrum Virtualize for Public Cloud only, specify the **-node** parameter to change an existing **-username** and **-chapsecret** where the target has a user name or CHAP secret per initiator node.

```
chiscsistorageport -username superman -chapsecret batman -node node1 5
```

The result is changing the authentication of iSCSI sessions from initiator node node1.

chnode

Use the **chnode** / **chnodecanister** command to change the name that is assigned to a node or node canister as well as other options. You can then use the new name when running subsequent commands. All parameters that are associated with this command are optional. However, you must specify one or more parameters.

Syntax

```
►►─ chnode ─ | chnodecanister ─┬───────────────────────┬─┬─────────────┬─ ──────────────►
                               ├─ -iscsialias ─ alias ─┤ └─ -failover ─┘
                               └─ -noiscsialias ───────┘

►─┬────────────────────────────────────────┬─ ─┬─────────────────────────┬─ ──────────────►
  └─ -name ─ new_node_or_nodecanister_name ─┘  └─ -identify ─┬─ yes ─┬─┘
                                                              └─ no ──┘
```

```
►─┬────────────────────────────┬── ─┬─ object_id ───┬──────────────────────────────────────────►◄
  ├─ -site ──┬─ site_id ───┬───┤    └─ object_name ─┘
  │          └─ site_name ─┘   │
  └─ -nosite ──────────────────┘
```

Parameters

-iscsialias *alias*
(Optional) Specifies the iSCSI name of the node or node canister. The maximum length is 79 characters. Do not use spaces for the iSCSI alias name.

Important: You can specify this parameter for online spares nodes.

-noiscsialias
(Optional) Clears any previously set iSCSI name for this node or node canister. This parameter cannot be specified with the **iscsialias** parameter.

Important: You can specify this parameter for online spares nodes.

-failover
(Optional) Specifies that the name or iSCSI alias being set is the name or alias of the partner node or node canister in the I/O group. When there is no partner node or node canister, the values set are applied to the partner node or node canister when it is added to the clustered system (system). If this parameter is used when there is a partner node or node canister, the name or alias of that node or node canister changes.

Important: You can specify this parameter for online spares nodes.

-name *new_node_or_nodecanister_name*
(Optional) Specifies the name to assign to the node or node canister.

Note: Node or node canister names supplied with -name on **chnode** / **chnodecanister** commands must not be in use already as node or node canister names or as node or node canister failover names.

Important: The iSCSI Qualified Name (IQN) for each node or node canister is generated using the clustered system and node or node canister names. If you are using the iSCSI protocol, changing either name also changes the IQN of all of the nodes or node canisters in the clustered system and might require reconfiguration of all iSCSI-attached hosts.

-identify *yes* | *no*
(Optional) Allows you to control the light-emitting diode (LED) used on the node. The values are *yes* or *no*.

Important: You can specify this parameter for online spares nodes.

-site *site_id* | *site_name*
(Optional) Specifies the numeric site value or site name for the existing node. The value is 1 or 2.

Note: The site assigned to the node cannot be changed if the system topology is HyperSwap or stretched.

-nosite
(Optional) Resets the site value.

object_id | *object_name*
(Required) Specifies the object name or ID that you want to modify. The variable that follows the parameter is either:

- The object name that you assigned when you added the node to the clustered system

- The object ID that is assigned to the node (not the worldwide node name)

Description

If the **failover** parameter is not specified, this command changes the name or iSCSI alias of the node or node canister. The name can then be used to identify the node or node canister in subsequent commands.

The **failover** parameter is used to specify values that are normally applied to the partner node or node canister in the I/O group. When the partner node or node canister is offline, the iSCSI alias and IQN are assigned to the remaining node or node canister in the I/O Group. The iSCSI host data access is then preserved. If the partner node or node canister is offline when these parameters are set, the node or node canister they are set on handles iSCSI I/O requests to the iSCSI alias specified, or the IQN that is created using the node or node canister name. If the partner node or node canister in the I/O group is online when these parameters are set, the partner node or node canister handles iSCSI requests to the iSCSI alias specified, and its node or node canister name and IQN change.

To change the name of the node (with I/O running) :

1. Make sure the host system has active sessions with both node canisters in the I/O group (hosting the volume on which the I/O occurs).
2. Change name of one node canister using **chnode** command.
3. From the host system, log out of the node canister whose name changes.
4. Rediscover the target iSCSI qualified name (IQN) from the host using the host operating system's discovery mechanism.
5. Login with the new target IQN discovered on the host system, and make sure the login succeeds.
6. Repeat steps 2-5 with the other node canister.

Note: When using VMware ESX, delete the static paths (in the iSCSI initiator properties) that contain the old target IQN.
This ensures that the node canister name change does not impact iSCSI I/O during events such as a target failover.

An invocation example

```
chnode -name newname -identify yes node8
```

The resulting output:

```
No feedback
```

An invocation example

```
chnode -name testnodeone nodeone
```

The resulting output:

```
No feedback
```

An invocation example

```
chnodecanister -name testnodeone nodeone
```

The resulting output:

```
No feedback
```

An invocation example

```
chnode -site 1 node2
```

The resulting output:

```
No feedback
```

An invocation example

```
chnodecanister -site 1 node2
```

The resulting output:

```
No feedback
```

chnodebattery

Use the **chnodebattery** command to set or clear the light-emitting diode (LED) on a hot-swappable battery (in a node). This command applies to SAN Volume Controller 2145-DH8 systems.

Syntax

```
►►─chnodebattery─ ─┬──────────────────────┬─ ─┬─────────┬─ ─┬────────────────────────┬─ ──────────►
                   └─-identify──┬─yes─┬─┘    └─-remove─┘    └─-battery──battery_id─┘
                                └─no──┘

►─┬──────┬─────────────────────────────────────────────────────────────────────────────────────►◄
  └─node─┘
```

Parameters

-identify
(Optional) Allows you to control the light-emitting diode (LED).

-battery *battery_id*
(Optional) Specifies the battery that is in the node.

-remove
(Optional) Specifies battery removal and terminates any calibration that runs on another battery.

node
(Optional) Specifies the node that the battery is in.

Description

This command notifies the battery back-up (BBU) driver that a user wants to remove a battery.

An invocation example to make the fault LED flash on battery 1 in node 3

```
chnodebattery -identify yes -battery 1 3
```

The resulting output:

```
No feedback
```

An invocation example to remove battery 1 in node 3

```
chnodebattery -remove -battery 1 3
```

The resulting output:

```
No feedback
```

chnodebootdrive

Use the **chnodebootdrive** command to change a drive or synchronize the drives on a system if a drive or field-replaceable unit (FRU) replacement drive breaks. This command applies to SAN Volume Controller 2145-DH8 systems.

Syntax

```
►►─chnodebootdrive─┬─────────────────────────────────────┬─┬─node_id───┬──────────────────►◄
                   ├─-sync──-force───────────────────────┤ └─node_name─┘
                   ├─-identify──┬─yes─┬──-slot── slot_id─┤
                   │            └─no──┘                  │
                   └─-resetleds──────────────────────────┘
```

Parameters

-sync
(Optional) Specifies synchronization of drives marked `can_sync`.

-force
(Optional) Forces synchronization (though taking the node offline might cause a volume to go offline).

Important: Using the force parameter might result in a loss of access. Use it only under the direction of your product support information.

-identify *yes* | *no*
(Optional) Controls the operation of the light-emitting diode (LED) of the drive in the specified slot.

-slot *slot_id*
(Optional) Specifies the boot drive slot. It must be used with the **-identify** parameter.

resetleds
(Optional) Clears the identify LEDs of all drives in the specified node and indicates `-identify no` is specified.

node_id | *node_name*
(Optional) Specifies the ID or name of the node.

Description

The command identifies and synchronizes drive information for system drives.

Specifying **-sync** causes a node restart on the specified node. This restart is not successful if any volume depends on that node.

Important: If **-force** is also specified, the system does not check for dependent volumes.

An invocation example

```
chnodebootdrive
```

The following output is displayed:

```
No feedback
```

An invocation example

```
chnodebootdrive -identify yes -slot 1 1
```

The following output is displayed:

```
No feedback
```

chnodehw (SVC) / **chnodecanisterhw** (Storwize family products)

Use the **chnodehw** / **chnodecanisterhw** command to update the hardware configuration for a node or node canister.

Syntax

```
►►── chnodehw ── | chnodecanisterhw ──┬─────────────────────┬──┬──────────┬── ──┬─ object_id ───┬──────►
                                      └─ -legacy ── version ─┘  └─ -force ─┘     └─ object_name ─┘

►─ ────────────────────────────────────────────────────────────────────────────────────────────────►◄
```

Parameters

-legacy *version*
(Optional) Sets the hardware configuration to make it compatible with the 6.3.0.0 code level. The format is four decimal numbers that are separated by periods, and there can be up to 16 characters.

-force
(Optional) Allow the node to restart and change its hardware configuration even if it causes volumes to go offline.

Important: Using the force parameter might result in a loss of access. Use it only under the direction of your product support information.

object_id | *object_name*
(Optional) Specifies the object name or ID.

Description

This command automatically reboots the node or node canister if the node or node canister hardware is different than its configured hardware. After rebooting, the node or node canister uses its hardware, and does not use the previous configuration.

Attention: When you run the **chnodehw** command to change the configured hardware for a node:

- Small Computer System Interface-3 (SCSI-3) reservations (through that node) are removed.
- Small Computer System Interface-3 (SCSI-3) registrations (through that node) are removed.

Note: This command fails if you remove the last compression card of a node and try to commit that change while compressed volumes still exist in that I/O group.

Use the **-legacy** parameter if you want to establish a partnership with another clustered system that is running an earlier level of code than the local system. The value that is supplied for the **-legacy** parameter must be the code level of the other clustered system.

An invocation example of how to update the node hardware configuration of node ID 7

```
chnodehw 7
```

The resulting output:

```
No feedback
```

An invocation example of how to update the node hardware configuration for the node named `node7` (including if the node reboot causes an I/O outage)

```
chnodehw -force node7
```

The resulting output:

```
No feedback
```

An invocation example of how to update the node hardware configuration for compatibility with code level `6.3.0.0`

```
chnodehw -legacy 6.3.0.0 node2
```

The resulting output:

```
No feedback
```

An invocation example of how to update the node canister hardware configuration of canister ID `7`

```
chnodecanisterhw 7
```

The resulting output:

```
No feedback
```

An invocation example of how to update the node canister hardware configuration for `canister7` (including if the canister reboot causes an I/O outage)

```
chnodecanisterhw -force canister7
```

The resulting output:

```
No feedback
```

chquorum

Use the **chquorum** command to change the quorum association.

Syntax

```
►►— chquorum — ┬──────────┬─ ┬────────────────────────────┬─ ┬─────────────────────┬─ ──────────►
               └─ -active ┘  ├─ -mdisk ──┬─mdisk_id───┬───┤  └─ -override ─yes|no─┘
                             │           └─mdisk_name─┘   │
                             └─ -drive ── drive_id ───────┘

►— quorum_id ─────────────────────────────────────────────────────────────────────────────────►◄
```

Parameters

-active
(Optional) Makes the specified quorum ID the active one. The **active** parameter must be used if neither the **mdisk** nor the **drive** parameters are specified.

-mdisk *mdisk_id* | *mdisk_name* | **-drive** *drive_id*
(Optional) Specifies the MDisk or drive to be this quorum ID.

Note: SAN Volume Controller systems use MDisks only.

-override *yes|no*

Enables the automatic quorum selection to be overridden. In this state, the quorum disk is only moved if the resources are offline. Do not use this parameter unless a specific quorum disk is required for the configuration.

quorum_id

(Required) Specifies which quorum ID to change. Permitted values are values are `0`, `1`, and `2`.

Description

Use the **chquorum** command to change the quorum association. To identify the drive or MDisk that is the current active quorum disk, use the **lsquorum** command.

Remember: You cannot use this command to change the active quorum device when you use an IP quorum application. To change the active IP quorum application, the quorum application must be restarted. The quorum application that connects first is chosen and is active (if valid).

The **chquorum** command is not synchronous, but usually takes only a few seconds to complete. In some situations it can take several minutes.

The clustered system (system) uses the quorum disk or drive as a tie breaker when exactly half of the nodes that were previously a member of the system are present.

Attention: Only assign quorum disks to drives in the control enclosure or to external MDisks. Some maintenance procedures require that quorum is moved temporarily to expansion enclosures. Once that procedure is complete, return the quorum drives to the control enclosure.

The use of a quorum disk or drive allows the system to manage a SAN fault that splits the system exactly in half. One half of the system continues to operate and the other half stops until SAN connectivity is restored.

There is only one quorum disk or drive; however, the system uses three as quorum candidates. The system selects the actual quorum disk or drive from the pool of quorum candidates. The quorum candidates also hold a copy of important system metadata. Just over 256 MB is reserved for this purpose on each quorum candidate disk. When using an MDisk as quorum disk, this space is allocated from the storage pool.

The number of extents required depends on the extent size for the storage pool containing the MDisk. Table 26 provides the number of extents reserved for quorum use by extent size.

Table 26. Number of extents reserved by extent size

Extent size (MB)	Number of extents reserved for quorum use
16	17
32	9
64	5
128	3
256	2
512	1
1024	1
2048	1
4096	1
8192	1

When you issue this command, the MDisk or drive that currently is assigned the quorum index number is set to a nonquorum disk. The system automatically assigns quorum indexes.

You can set the active quorum disk or drive with the **-active** parameter. This can be useful in a system configuration to ensure that the most highly-available quorum disk or drive is used.

Note: Quorum disks must be allocated one per site when the system topology is `stretched` or `hyperswap`.

An invocation example

```
chquorum -mdisk 45 2
```

The resulting output:

```
No feedback
```

chsecurity

2 Use the **chsecurity** command to change the Secure Sockets Layer (SSL), Secure Shell (SSH), or Transport
Layer Security (TLS) security settings for a system.

Syntax

```
►►—chsecurity— ——┬-sslprotocol—security_level—┬———————————————————————————————————————►◄
                 └-sshprotocol—security_level—┘
```

Parameters

Remember: These parameters are mutually exclusive. You must specify **-sslprotocol** or **-sshprotocol**, not both.

-sslprotocol *security_level*

1 (Required) Specifies the numeric value for the SSL security level setting, which can take any value
1 from 1 to 4. A setting of 3 is the default value.

A security level setting of:

- 1 disallows SSL 3.0.
- 2 allows TLS 1.2 only.
- 3 additionally disallows TLS 1.2 cipher suites that are not exclusive to 1.2.
- 2 4 additionally disallows RSA key exchange ciphers.

-sshprotocol *security_level*

2 (Required) Specifies the numeric value for the SSH security level setting, which can take a value of 1
2 or 2. A setting of 1 is the default value.

2 A security level setting of:

- 2 1 allows the following key exchange methods:
  - 2 curve25519-sha256
  - 2 curve25519-sha256@libssh.org
  - 2 ecdh-sha2-nistp256
  - 2 ecdh-sha2-nistp384
  - 2 ecdh-sha2-nistp521
  - 2 diffie-hellman-group-exchange-sha256
  - 2 diffie-hellman-group16-sha512
  - 2 diffie-hellman-group18-sha512

2 – diffie-hellman-group14-sha256

2 – diffie-hellman-group14-sha1

2 – diffie-hellman-group1-sha1

2 – diffie-hellman-group-exchange-sha1

2 • 2 allows the following key exchange methods:

2 – curve25519-sha256

2 – curve25519-sha256@libssh.org

2 – ecdh-sha2-nistp256

2 – ecdh-sha2-nistp384

2 – ecdh-sha2-nistp521

2 – diffie-hellman-group-exchange-sha256

2 – diffie-hellman-group16-sha512

2 – diffie-hellman-group18-sha512

2 – diffie-hellman-group14-sha256

2 – diffie-hellman-group14-sha1

Description

2 This command changes the SSL, SSH, or TLS security settings on a system.

Important: If you use SSL or TLS, changing the security could disrupt these services.

If this occurs:

1. Wait 5 minutes and try again. (Wait for any services to restart.)
2. Confirm that the SSL or TLS implementation is up-to-date and supports the specified level of security.
3. If necessary, revert to an earlier version of SSL or TLS security.

2 An invocation example

```
2 chsecurity -sslprotocol 4
```

2 The resulting output:

```
2 Changing the SSL security level could disable the GUI connection on old web browsers,
2 and changing the SSH security level may logout existing SSH sessions. Are you sure you want to continue? (y/yes to confirm)
```

2 An invocation example

```
2 chsecurity -sshprotocol 2
```

2 The resulting output:

```
2 Changing the SSL security level could disable the GUI connection on old web browsers,
2 and changing the SSH security level may logout existing SSH sessions. Are you sure you want to continue? (y/yes to confirm)
```

chsite

Use the **chsite** command to change the site name.

Syntax

```
►►─chsite─ ─-name─new_ site_name─ ──┬─site_id────────────┬────────────────────────────────────►◄
                                    └─existing_site_name─┘
```

Parameters

-name *new_site_name*
(Required) Specifies the new name for the site.

site_id | *existing_site_name*
(Required) Specifies the existing site ID or site name that is being changed.

Description

This command changes the site name.

Remember: This command is only applicable when a system is configured as a stretched system or a HyperSwap system (by using the **chsystem -topology** command).
In a stretched configuration these applications are spread across two or more geographic locations or sites:

- Nodes
- Storage
- Host servers
- Infrastructure

An invocation example

```
chsite -name Quorum 3
```

The resulting output:

```
No feedback
```

chsra

Use the **chsra** command to configure support assistance.

Syntax

```
►►── chsra ── ─┬───────────────┬─ ─┬─────────────────────────────────┬──────────────►
               ├─ -enable ─────┤   └─ -remotesupport ──┬─ enable ──┬─┘
               ├─ -disable ────┤                       ├─ disable ─┤
               └─ -updatetoken ┘                       └─ test ────┘

►─┬──────────────────────────────────────┬──────────────────────────────────────►◄
  └─ -idletimeout ── timeout_in_minutes ─┘
```

Parameters

-enable
(Optional) Creates remote access accounts and enables local support assistance.

-disable
(Optional) Deletes all remote access accounts and disables local and remote support assistance.

-updatetoken
(Optional) Updates the shared security token that is used for support assistance.

-remotesupport *enable* | *disable* | *test*
(Optional) Configures remote support assistance directly over the internet or by a configured proxy server. The values are:

- enable

- `disable`
- `test`

No default value exists.

-idletimout *timeout_in_minutes*
(Optional) Enables remote support for a limited amount of time (specified in minutes). The value must be a positive number (integer) denoting how many minutes remote support assistance is idle (and timed out). This parameter does not time out when a support session is in progress on any of your system nodes. It is renewed as many times as needed and only times out after all active sessions are terminated.

Remember: If the idle timeout expires on all participating nodes in a system, remote system support is disabled. If remote system support is disabled, remote system support is not automatically started on events that include:

- When new nodes join the system.
- T3 recovery procedures.
- Node warm or cold starts.

If you specify **-idletimout**, you must also specify **-remotesupport**.

Description

This command configures local or remote support assistance.

Note: Turn on both local and remote support assistance to more efficiently resolve any problems that are encountered.

Remote support assistance is available either directly over the internet or by using a proxy server. Remote system support is routed by using the proxy server if any proxy servers are configured. You must do the following to turn on remote support assistance:

1. Configure service IP on all system nodes.
2. Configure call home and heartbeat functions on the system.
3. Configure local support assistance on your system, which creates support and sets up authentication. For storage systems that have direct access to the internet, the firewall must allow inbound and outbound connections to Internet Protocol (IP) addresses `129.33.206.139` and `204.146.30.139` on port `22`. If you must use a proxy server, configure it by using the **mksystemsupportcenter** command.

An invocation example for creating support assistance accounts and enabling local support assistance

```
chsra -enable
```

The detailed resulting output:

```
No feedback
```

An invocation example for deleting support user accounts and disabling local support assistance

```
chsra -disable
```

The detailed resulting output:

```
No feedback
```

An invocation example for updating the shared token that is used for challenge response authentication

```
chsra -updatetoken
```

The detailed resulting output:

```
No feedback
```

An invocation example for enabling remote support assistance

```
chsra -remotesupport enable
```

The detailed resulting output:

```
No feedback
```

An invocation example for enabling remote support assistance for 30 minutes

```
chsra -remotesupport enable -idletimeout 30
```

The detailed resulting output:

```
No feedback
```

An invocation example to test remote support assistance (which is not enabled after test completion)

```
   chsra -remotesupport test
```

The detailed resulting output:

```
No feedback
```

chsystem

Use the **chsystem** command to modify the attributes of an existing clustered system (system). Enter this command any time after a system is created. All the parameters that are associated with this command are optional. However, you must specify one or more parameters with this command.

Syntax

```
►►── chsystem ── ─┬──────────────────────────┬─ ─┬──────────────────────────────────────┬─ ── ──────────►
                  └─ -name ── system_name ───┘   └─ -rcbuffersize ── new_size_in_MB ───┘

►─┬──────────────────────┬─ ─┬──────────────────────────────────────┬─ ─────────────────────►
  └─ -alias ── id_alias ─┘   └─ -icatip ── icat_console_ip_address ─┘

►─┬───────────────────────────────────┬─ ─┬──────────────────────────────────────┬─ ───────────►
  └─ -invemailinterval ── interval ───┘   └─ -gmlinktolerance ── link_tolerance ─┘

►─┬─────────────────────────────────────┬─ ─┬───────────────────────────────────┬─ ───────────►
  └─ -gmmaxhostdelay ── max_host_delay ─┘   └─ -icatip ── ipv4_icat_ip_address ─┘

►─┬─────────────────────────────────────┬─ ─┬─────────────────────────────────┬─ ─────────────►
  └─ -icatip_6 ── ipv6_icat_ip_address ─┘   └─ -ntpip ── ipv4_ntp_ip_address ─┘

►─┬───────────────────────────────────┬─ ─┬─────────────────────────────────┬─ ───────────────►
  └─ -ntpip_6 ── ipv6_ntp_ip_address ─┘   └─ -isnsip ── sns_server_address ─┘

►─┬────────────────────────────────────────┬─ ────────────────────────────────────────────────►
  └─ -isnsip_6 ── ipv6_sns_server_address ─┘
```

```
►─┬───────────────────────────────────────────────────┬─ ─┬────────────────────────┬─ ──────────►
  └─ -relationshipbandwidthlimit ── bandwidth_in_mBps ─┘   └─ -infocenterurl ── url ─┘

►─┬─────────────────────────────────────────────────────┬──────────────────────────────────────────►
  └─-iscsiauthmethod──┬─none──────────────────────────────┤
                      └─chap──-chapsecret── chap_secret ─┘

►─┬────────────────────────────────────────────────────┬┬──────────────────────────┬─ ──────────►
  └─ -rcauthmethod ──┬─none───────────────────────────────┤├─-chapsecret── chap_secret ─┤
                     └─chap──-chapsecret── chap_secret ─┘└─-nochapsecret──────────────┘

►─┬──────────────────────────────────────────────────────────────┬─ ─┬──────────────────────────┬──►
  └─-iscsiusername────iscsi username for two way authentication───┘   └─-layer──┬─replication─┤
                                                                                 └─storage─────┘

►─ ─┬──────────────────────────────┬─ ─ ─┬──────────────────────────────────┬─ ──────────────────►
    └─ -cacheprefetch ──┬─ on ───┤       └─ -localfcportmask ──port_mask──┘
                        └─ off ─┘

►─┬──────────────────────────────────────────┬─ ─┬────────────────────────────────┬─ ─────────────►
  └─ -partnerfcportmask ──port_mask──┘           └─ -hightempmode ──┬─on──┤
                                                                     └─off─┘

►─┬──────────────────────────────────┬─ ─┬──────────────────────────────────────────────────┬─ ───►
  └─ -topology ──┬─standard──┤           └─ -vdiskprotectiontime ── value_in_minutes ─┘
                 ├─stretched─┤
                 └─hyperswap─┘

►─┬──────────────────────────────────────────────┬─ ─┬──────────────────────┬─ ──────────────────►
  └─ -vdiskprotectionenabled ──┬─ yes ──┤            └─ -odx ──┬─ on ───┤
                               └─ no ──┘                       └─ off ─┘

►─┬──────────────────────────────────────────┬─ ─┬────────────────────────────────────────────────┬─ ─►
  └─ -easytieracceleration ──┬─ on ───┤          └─ -maxreplicationdelay ── value_in_seconds ─┘
                             └─ off ─┘

►─┬──────────────────────────────────────────────────────────┬─ ─ ─┬──────────────────────────────┬►
  └─ -partnershipexclusionthreshold ── value_in_seconds ─┘         └─ -ibmcustomer ── customer_id ─┘

►─┬──────────────────────────────────────┬┬────────────────────────────────┬─ ─────────────────────►
  └─ -ibmcomponent ── component_id ─┘    └─ -ibmcountry ── country_id ─┘

►─┬──────────────────────────────────────┬─ ─┬──────────────────────────────────┬─ ──────────────►
  └─ -enhancedcallhome ──┬─ on ───┤          └─ -censorcallhome ──┬─ on ───┤
                         └─ off ─┘                                └─ off ─┘

►─┬─────────────────────────┬─────────────────────────────────────────────────────────────────────►◄
  └─ -unmap ──┬─ on ───┤
              └─ off ─┘
```

Parameters

-name *system_name*
(Optional) Specifies a new name for the system.

Important: The Internet Small Computer System Interface (iSCSI) Qualified Name (IQN) for each node is generated by using the system and node names. If you are using the iSCSI protocol, changing either name also changes the IQN of all of the nodes in the system and might require reconfiguration of all iSCSI-attached hosts.

-rcbuffersize *new_size_in_MB*
(Optional) Specifies the amount of memory, in megabytes (MB), to use on each node for Metro Mirror and Global Mirror communications, from 48 to 512 MB. The default is 48 MB.

Important: Adjust this setting only when directed by your support team.

All nodes in the system must be online and have a minimum of 8 gigabytes (GB) - or 8192 megabytes (MB) - of memory to change this setting.

Remember:

- Before changing this setting you must stop all partnerships with this system.
- This parameter operates on the local system only and changing it is disruptive to mirroring operations.

-alias *id_alias*
(Optional) Specifies an alternative name that does not change the basic ID for the system, but does influence the `VDisk_UID` of every `vdiskhostmap`, both existing and new. These objects are created for a system whose ID matches the alias. Therefore, changing the system alias causes loss of host system access until each host scans for volumes that are presented by the system.

-invemailinterval *interval*
(Optional) Specifies the interval at which inventory emails are sent to the designated email recipients. The interval range is `0` to `15`. The interval is measured in days. Setting the value to 0 turns off the inventory email notification function.

-gmlinktolerance *link_tolerance*
(Optional) Specifies the length of time, in seconds, for which an inadequate intersystem link is tolerated for a Global Mirror operation. The parameter accepts values from `20` to `86400` seconds in steps of 10 seconds. The default is 300 seconds. You can disable the link tolerance by entering a value of `0` for this parameter.

-gmmaxhostdelay *max_host_delay*
(Optional) Specifies the maximum time delay, in milliseconds, at which the Global Mirror link tolerance timer starts counting down. This threshold value determines the additional impact that Global Mirror operations can add to the response times of the Global Mirror source volumes. You can use this parameter to increase the threshold from the default value of 5 milliseconds.

-icatip *icat_console_ip_address*
(Optional) Specifies the system's new IPv4 address that is used by the system. The format of this IP address must be a dotted decimal notation with the port; for example, `255.255.255.255:8080`. If you specify this parameter, it overwrites any existing **-icatip_6** address.

-icatip_6 *icat_console_ipv6_address*
(Optional) Specifies the system's new IPv6 address. If you specify this parameter, it overwrites any existing **-icatip** address. The format of the IPv6 address must be:

- Eight colon-separated groups of four hexadecimal digits; for example:

  `[1234:1234:abcd:0123:0000:0000:7689:6576]:23`
- Eight colon-separated groups of hexadecimal digits with leading zeros omitted; for example:

  `[1234:1234:abcd:123:0:0:7689:6576]:23`
- Suppression of one or more consecutive all 0 groups; for example:

  `[1234:1234:abcd:123::7689:6576]:23`

-ntpip *ipv4_ntp_ip_address*
(Optional) Specifies the IPv4 address for the Network Time Protocol (NTP) server. Configuring an NTP server address causes the system to use that NTP server as its time source. Specify the **-ntpip** parameter with a zero address to use another time source:

```
chsystem -ntpip 0.0.0.0
```

-ntpip_6 *ipv6_ntp_ip_address*

Note: Before you specify **-ntpip_6**, an IPv6 prefix and gateway must be set for the system. (Optional) Specifies the IPv6 address for the NTP server. Configuring an NTP server address causes the system to immediately start using that NTP server as its time source. To choose another time source, specify the **-ntpip_6** parameter with a zero address, as follows:

```
chsystem -ntpip_6 0::0
```

-isnsip *sns_server_address*
(Optional) Specifies the IPv4 address for the iSCSI storage name service (SNS). Specify the **-isnsip** parameter with a zero address to select another IPv4 iSCSI SNS server:

```
chsystem -isnsip 0.0.0.0
```

-isnsip_6 *ipv6_sns_server_address*
(Optional) Specifies the IPv6 address for the iSCSI SNS. Specify the **-isnsip_6** parameter with a zero address to select another configured IPv6 iSCSI SNS server:

```
chsystem -isnsip_6 0::0
```

-relationshipbandwidthlimit *bandwidth_in_mBps*
(Optional) Specifies the new background copy bandwidth in megabytes per second (MBps), from 1 to 1000. The default is 25 MBps.

Important: For partnerships over IP links with compression, this parameter specifies the aggregate bandwidth after compression was applied to the data. Do not set this parameter higher than the physical link bandwidth multiplied by the (carefully rounded down) compression factor.

This parameter operates system-wide and defines the maximum background copy bandwidth that any relationship can adopt. The existing background copy bandwidth settings that are defined on a partnership continue to operate, with the lower of the partnership and volume rates attempted.

Note: Do not set this value higher than the default without establishing that the higher bandwidth can be sustained.

-infocenterurl *url*
Specifies the preferred online documentation URL to override the one used by the GUI. Because this information is interpreted by the Internet browser, the specified information might contain a hostname or an IP address.

Remember: View the currently configured URL in the GUI preferences window. This window can also help reset this value to the default setting.

-iscsiauthmethod none | chap -chapsecret *chap_secret*
(Optional) Sets the authentication method for the iSCSI communications of the system:

- chap indicates Internet Small Computer System Interface (iSCSI) authentication is turned on.

  Remember: This turns on iSCSI partnership authentication when a Challenge Handshake Authentication Protocol (CHAP) secret key is set for the system.
- none indicates that iSCSI partnership authentication is turned off.

-rcauthmethod none | chap -chapsecret *chap_secret*
(Optional) Turns authentication on or off for remote copy partnership requests that are native IP partnerships. Remote copy includes Metro Mirror, Global Mirror, and HyperSwap. Additionally:

- chap indicates that remote copy authentication is turned on.

  Remember: This action turns on authentication of remote copy partnership requests when a Challenge Handshake Authentication Protocol (CHAP) secret key is set for the system.
- none indicates that remote copy partnership authentication is turned off.

-chapsecret*chap_secret*
(Optional) Sets the CHAP secret to be used to authenticate the system that uses iSCSI. This parameter is required if the **iscsiauthmethod chap** parameter is specified. The specified CHAP secret cannot begin or end with a space.

-nochapsecret
(Optional) Clears any previously set CHAP secret for iSCSI authentication. The **nochapsecret** parameter cannot be specified when **chapsecret** is specified.

-iscsiusername
(Optional) Specifies the user name for the entire SVC system that is used for two-way authentication for iSCSI host attach login. If this parameter is specified, this value is taken as "username" for two-way authentication in iSCSI host attach login. If user name is not specified, user name for two-way chap authentication is NULL.

-layer *replication* | *storage*
(Optional) Specifies which layer a system is in. The system can create partnerships with systems in the same layer.

Note: If you specify **-layer** you must specify either *replication* or *storage*. This option can be used if no other systems are visible on the fabric, and no system partnerships are defined.

-cacheprefetch *on* | *off*
(Optional) Indicates whether cache prefetching is enabled or disabled across the system. Adjust this only when following direction from your product support information.

-localfcportmask *port_mask*
(Optional) Indicates the Fibre Channel (FC) input/output (I/O) port mask for local system node-to-node communications only. Fibre Channel port mask does not affect host or storage system traffic. The port mask is 64 binary bits and is made up of a combination of 0's and 1's, where 0 indicates that the corresponding FC I/O port cannot be used and 1 indicates that it can be used. The mask is applied to all nodes in the local system. At least two ports must be selected for local system node-to-node communications. The mask must result in at least 2 FC connections between each node in the local system, using only the selected ports and FC zones visible to those ports. Valid mask values might range from 0011 (only ports 1 and 2 enabled) to 1111111111111111111111111111111111111111111111111111111111111111 (all ports enabled). For example, a mask of 111111101101 enables ports 1, 3, 4, 6, 7, 8, 9, 10, 11, and 12.

Remember: A partial mask (fewer than 64 characters) is zero-extended, meaning that any ports that are not specified are not enabled.
Specify the **lsportfc** command to display FC I/O port IDs.

-partnerfcportmask *port_mask*
(Optional) Indicates the FC I/O port mask for partnered system-to-system communications only. Fibre Channel port mask does not affect host or storage system traffic. The port mask is 64 binary bits and is made up of a combination of 0's and 1's, where 0 indicates that the corresponding FC I/O port cannot be used and 1 indicates that it can be used. The mask is applied to all nodes in the local system. Valid mask values might range from 0000 (no ports enabled) to 1111111111111111111111111111111111111111111111111111111111111111 (all ports enabled). For example, a mask of 111111101101 enables ports 1, 3, 4, 6, 7, 8, 9, 10, 11, and 12.

Remember: A partial mask (fewer than 64 characters) is zero-extended, meaning that any ports not specified are not enabled.
Specify the **lsportfc** command to display FC I/O port IDs.

-hightempmode *on* | *off*
(Optional) Sets (or removes) High Temperature Mode (HTM). The values are on and off.

-topology *standard* | *stretched* | *hyperswap*
(Optional) Indicates the intended system topology, which is either *standard*, *stretched*, or *hyperswap*.

-vdiskprotectiontime *value_in_minutes*
(Optional) Sets volume protection time (in minutes).

-vdiskprotectionenabled *yes* | *no*
(Optional) Enables or disables volume protection. The values are yes and no.

-odx *on* | *off*
(Optional) Enables or disables offloaded data transfers (ODX). The values are on and off.

-easytieracceleration *on* | *off*
(Optional) Enables Easy Tier and pool balancing acceleration. The values are on and off.

-maxreplicationdelay *value_in_seconds*
(Optional) Sets a maximum replication delay in seconds. The value must be a number from 0 to 360.

-partnershipexclusionthreshold *value_in_seconds*
(Optional) Sets the timeout for an I/O operation (in seconds) for remote systems. The value must be a number from 30 to 315 (default).

-ibmcustomer *customer_id*
(Optional) Specifies the customer number assigned when a software license that is automatically added to the entitlement database. The value must be a number that contains 7 - 10 digits (or blank).

-ibmcomponent *component_id*
(Optional) Specifies the component ID used for entitlement and call home system. The value is blank or SANVCNSW1.

-ibmcountry *country_id*
(Optional) Specifies the country ID used for entitlement and call home system. The value is blank or a 3-digit number.

1 **-enhancedcallhome** *on* | *off*
1 (Optional) Specifies that the call home function is to send enhanced reports to the support center.
1 Valid values are yes or no.

1 The enhanced reports include operational and event-related data and specific configuration
1 information that is included in the inventory report. This function alerts the support center about
1 hardware failures and potentially serious configuration or environmental issues. The support center
1 can use the configuration information to automatically generate best practices or recommendations
1 that are based on your actual configuration.

1 **-censorcallhome** *on* | *off*
1 (Optional) Specifies that sensitive data is deleted from the enhanced call home data. The values are
1 yes or no.

-unmap *on* | *off*
(Optional) Specifies whether the system administrator enables the Small Computer System Interface (SCSI) unmap feature. The values are on (default) or off.

Description

This command modifies specific features of a system. Multiple features can be changed by issuing a single command.

Using the **-ntpip** or **-ntpip_6** parameter allows the system to use an NTP server as an outside time source. The system adjusts the system clock of the configuration node according to time values from the NTP server. The clocks of the other nodes are updated from the configuration node clock. In the NTP mode, the **setsystemtime** command is disabled.

All command parameters are optional, but you must specify at least one parameter.

Use the **chsystemip** command to modify the system IP address and service IP address.

Remember: Setting a CHAP secret key for the system does not turn on authentication for iSCSI hosts or remote copy partnerships. Remote copy includes Metro Mirror, Global Mirror, and HyperSwap. Turn off authentication by issuing **-iscsiauthmethod** or **-rciauthmethod**.

The topology can be set to HyperSwap only if node sites have been configured. All nodes must have a site defined. If both nodes are defined in an I/O group they must be assigned to the same site (either 1 or 2; 3 cannot be used for nodes). You must have at least one I/O group with both nodes in site 1 and at least one I/O group with both nodes in site 2.

Note: If there are any active relationships defined, the system topology must be HyperSwap.

An invocation example

```
chsystem -ntpip 9.20.165.16
```

The resulting output:

```
No feedback
```

An invocation example to set up an external NTP server

```
chsystem -ntpip 123.234.123.234
```

The resulting output:

```
No feedback
```

An invocation example to change the preferred `infocenterurl` value

```
chsystem -infocenterurl http://miscserver.company.com/ibm/infocenter
```

The resulting output:

```
No feedback
```

An invocation example to change the local port mask value

To set the local mask to sixty-two 0's and two 1's, indicating FC I/O ports with IDs 1 and 2 are capable of local node communication:

```
chsystem -localfcportmask 11
```

The resulting output:

```
No feedback
```

An invocation example to change the partner port mask value

To set the partner mask to sixty-three 0's and one 1, indicating that FC I/O port with ID 2 is capable of remote node communication:

```
svctask chsystem -partnerfcportmask 0010
```

The resulting output:

```
No feedback
```

An invocation example to change the HTM

```
chsystem -hightempmode on
```

The resulting output:

```
No feedback
```

An invocation example to set the system topology

```
chsystem -topology standard
```

The resulting output:

```
No feedback
```

An invocation example to set authentication for remote copy

```
chsystem -chapsecret ABCB1234 -iscsiauthmethod none -rcauthmethod chap
```

The resulting output:

```
No feedback
```

An invocation example to turn off volume protection

```
chsystem -vdiskprotectionenabled no
```

The resulting output:

```
No feedback
```

An invocation example to turn on volume protection and set the protection time to 60 minutes

```
chsystem -vdiskprotectionenabled yes -vdiskprotectiontime 60
```

The resulting output:

```
No feedback
```

An invocation example to turn on Easy Tier acceleration

```
chsystem -easytieracceleration on
```

The resulting output:

```
No feedback
```

An invocation example to turn on ODX

```
chsystem -odx on
```

The resulting output:

```
No feedback
```

An invocation example to set the maximum replication delay

```
chsystem -maxreplicationdelay 100
```

The resulting output:

```
No feedback
```

An invocation example to set the partnership exclusion threshold

```
chsystem -partnershipexclusionthreshold 120
```

The resulting output:

```
No feedback
```

An invocation example to specify an IBM customer ID, component ID, and country ID

```
chsystem -ibmcustomer 1928374 -ibmcomponent SANVCNSW1 -ibmcountry 001
```

The resulting output:

```
No feedback
```

1 An invocation example to turn off enhanced call home

```
1 chsystem -enhancedcallhome off
```

1 The resulting output:

```
1 No feedback
```

1 An invocation example to turn on censor call home

```
1 chsystem -censorcallhome on
```

1 The resulting output:

```
1 No feedback
```

chsystemcert

Use the **chsystemcert** command to manage the Secure Sockets Layer (SSL) certificate that is installed on a clustered system (system).

Syntax

```
►►— chsystemcert — ┬────────────────┬─ ┬─────────────────────┬─ ┬───────────────────┬─ ──────────►
                   └─ -mkselfsigned ─┘  └─ -country —country─┘  └─ -state —state─┘

►─┬─────────────────────────┬─ ┬──────────────────────────┬─ ┬──────────────────────────────────┬─ ──────────►
  └─ -locality —locality─┘     └─ -org —organization─┘       └─ -orgunit —organizationunit─┘

►─┬───────────────────┬─ ┬─────────────────────────────┬─ ┬───────────────────────┬─ ──────────►
  └─ -email —email─┘     └─ -commonname —commonname─┘     └─ -keytype —keytype─┘

►─┬─────────────────────┬─ ─────────────────────────────────────────────────────────►◄
  └─ -validity —days─┘


►►— chsystemcert — ┬──────────────┬─ — -country —country— — -state —state— ──────────────►
                   └─ -mkrequest ─┘

►— -locality —locality— — -org —organization— — -orgunit —organizationunit— ──────────────►

►— -email —email— — -commonname —commonname— ┬───────────────────────┬─ ┬──────────┬─ ──────►◄
                                             └─ -keytype —keytype─┘     └─ -force ─┘


►►— chsystemcert — ┬────────────┬─ ┬───────────────────────────────┬─ ────────────────►◄
                   └─ -install ─┘  └─ -file —input_file_pathname─┘


►►— chsystemcert — ┬───────────┬─ ────────────────────────────────────────────────►◄
                   └─ -export ─┘
```

Parameters

-mkselfsigned
(Optional) Generates a self-signed SSL certificate. If you do not specify **-mkselfsigned**, you must specify **-mkrequest**, **-export**, or **-install**.

-mkrequest
(Optional) Generates a certificate request. If you do not specify **-mkrequest**, you must specify **-mkselfsigned**, **-export**, or **-install**.

-country *country*
(Optional for **-mkselfsigned** and required for **-mkrequest**) Specifies the 2-digit country code for the self-signed certificate or certificate request.

-state *state*
(Optional for **-mkselfsigned** and required for **-mkrequest**) Specifies the state information for the self-signed certificate or certificate request. The value can be an ASCII string from 0 - 128 characters.

-locality *locality*
(Optional for **-mkselfsigned** and required for **-mkrequest**) Specifies the locality information for the self-signed certificate or certificate request. The value can be an ASCII string in the range 0 - 128 characters.

-org *organization*
(Optional for **-mkselfsigned** and required for **-mkrequest**) Specifies the organization information for the SSL certificate. The value can be an ASCII string in the range 0 - 64 characters.

-orgunit *organizationunit*
(Optional for **-mkselfsigned** and required for **-mkrequest**) Specifies the organization unit information for the SSL certificate. The value can be an ASCII string in the range 0 - 64 characters.

-email *email*
(Optional for **-mkselfsigned** and required for **-mkrequest**) Specifies the email address that is used in the SSL certificate. The value can be an ASCII string in the range 0 - 64 characters.

-commonname *commonname*
(Optional for **-mkselfsigned** and required for **-mkrequest**) Specifies the common name for the SSL certificate. The value can be an ASCII string of 0 - 64 characters.

-validity *days*
(Optional) Specifies the number of days (1-9000) that the self-signed certificate is valid.

-keytype *keytpye*
(Optional) Specifies the SSL certificate key type. The supported key types are:

- rsa2048
- ecdsa384
- ecdsa521

-install
(Optional) Installs a certificate. If you do not specify **-install**, you must specify **-mkselfsigned**, **-mkrequest**, or **-export**.

-file
(Optional) Specifies the absolute path name of the certificate to install.

-export
(Optional) Exports the current SSL certificate. The certificate is exported to the /dumps/certificate.pem directory on the configuration node. If you do not specify **-export**, you must specify **-mkselfsigned**, **-mkrequest**, or **-install**.

-force
(Optional) Specifies that the certificate request can be deleted.

Description

Use this command to manage the SSL certificate that is installed on a system. You can also do the following items.

- Generate a new self-signed SSL certificate.
- Create a certificate request to be copied from the system and signed by a certificate authority (CA).

  Note: The signed certificate that is returned by the CA can be installed.
- Export the current SSL certificate (for example to allow the certificate to be imported into a key server).

Important: You must specify one of the following parameters:

- **`-mkselfsigned`**
- **`-mkrequest`**
- **`-install`**
- **`-export`**

An invocation example to create a self-signed certificate

```
chsystemcert -mkselfsigned
```

The detailed resulting output:

```
No feedback
```

An invocation example to create a self-signed certificate with a common name

```
chsystemcert -mkselfsigned -commonname weiland.snpp.com
```

The detailed resulting output:

```
No feedback
```

An invocation example to create a self-signed certificate with a key type and a 1-year validity period

```
chsystemcert -mkselfsigned -keytype ecdsa521 -validity 365
```

The detailed resulting output:

```
No feedback
```

chsystemip

Use the **`chsystemip`** command to modify the Internet Protocol (IP) configuration parameters for the clustered system (system).

Syntax

```
►►── chsystemip ── ┬─────────────────────────┬─ ┬─────────────────┬─ ──────────────►
                   └─ -clusterip ── ipv4addr ┘  └─ -gw ── ipv4addr ┘

►─┬──────────────────────┬─ ┬───────────────────────────┬─ ┬──────────────────┬─ ────►
  └─ -mask ── subnet_mask┘  └─ -clusterip_6 ── ipv6addr ┘  └─ -gw_6 ── ipv6addr┘

►─┬─────────────────────┬─ ── -port ── system_port ── ─────────────────────────────►◄
  └─ -prefix_6 ── prefix┘
```

```
►►─ chsystemip ─ ──┬─────────┬── ──┬───────────┬── ──┬──────────────────────┬── ─────────────►◄
                   └─ -noip ─┘     └─ -noip_6 ─┘     └─ -port ─ system_port ─┘
```

Parameters

-clusterip *ipv4addr*
(Optional) Changes the IPv4 system IP address. When you specify a new IP address for a system, the existing communication with the system is broken.

Important: The **-clusterip** parameter cannot be used if there are any active IPv4 partnerships with the system.

-gw *ipv4addr*
(Optional) Changes the IPv4 default gateway IP address of the system.

-mask *subnet_mask*
(Optional) Changes the IPv4 subnet mask of the system.

-noip
(Optional) Unconfigures the IPv4 stack on the specified port, or both ports if none is specified.

Note: This parameter does not affect node service address configurations.

-clusterip_6 *ipv6addr*
(Optional) Sets the IPv6 system address for the port.

Important: The **-clusterip_6** parameter cannot be used if there are any active IPv6 partnerships with the system.

-gw_6 *ipv6addr*
(Optional) Sets the IPv6 default gateway address for the port.

-prefix_6 *prefix*
(Optional) Sets the IPv6 prefix.

-noip_6
(Optional) Unconfigures the IPv6 stack on the specified port, or both ports if none is specified.

Note: This parameter does not affect node service address configurations.

-port *system_port*
(Required) Specifies which port (1 or 2) to apply changes to. This parameter is required unless the **noip** or **noip_6** parameter is used.

Description

This command modifies IP configuration parameters for the system. The first time you configure a second port, all IP information is required. Port 1 on the system must always have one stack fully configured.

There are two active system ports on the configuration node. There are also two active service ports on any node in which you are performing a service action.

If the system IP address is changed, the open command-line shell closes during the processing of the command. You must reconnect to the new IP address if connected through that port.

If there is no port 2 available on any of the system nodes, the **chsystemip** command fails.

The **noip** and **noip_6** parameters can be specified together only if the **port** is also specified. The **noip** and **noip_6** parameters cannot be specified with any parameters other than **port**.

Note: The **noip** and **noip_6** parameters do not affect node service address configurations. Port 1 must have an IPv4 or IPv6 system address. The configuration of port 2 is optional.

Service IP addresses for all ports and stacks are initialized to Dynamic Host Configuration Protocol (DHCP). A service IP address is always configured.

Note: If the `console_ip` is the same as IP address system port 1, Internet Protocol Version 4 (IPv4) followed by IPv6, change the `console_ip` when the system IP is changed. If the `console_ip` differs from the system port 1 IP address, do not change the `console_ip` when the system IP is changed.

To modify an IP address, list the IP address of the system by issuing the **lssystem** command. Modify the IP address by issuing the **chsystemip** command. You can either specify a static IP address or have the system assign a dynamic IP address.

Table 27 provides IP address formats that are supported.

Table 27. IP address list formats

IP type	IP address list format
IPv4	`1.2.3.4`
Full IPv6	`1234:1234:abcd:0123:0000:0000:7689:6576`
Full IPv6, leading zeros suppressed	`1234:1234:abcd:123:0:0:7689:6576`
IPv6 with zero compression	`1234:1234:abcd:123::7689:6576`

An invocation example

```
chsystemip -clusterip 9.20.136.5 -gw 9.20.136.1 -mask 255.255.255.0 -port 1
```

The resulting output:

```
No feedback
```

An invocation example

```
chsystemip -clusterip_6 2001:0db8:85a3:0000:0000:8a2e:0370:7334 -gw_6 2001:0db8:85a3:0000:0000:8a2e:0370:7334 -prefix_6 64 -port 2
```

The resulting output:

```
No feedback
```

chthrottle

Use the **chthrottle** command to change attributes associated with a specified throttle object.

Syntax

```
►►── chthrottle ──┬────────────────────────────────────┬── ┬──────────────────────┬── ──────────►
                  └─ -bandwidth ── bandwidth_limit_in_mb ─┘  └─ -iops ── iops_limit ─┘

►─┬──────────────────────┬── ──┬─ throttle_id ───┬──────────────────────────────────────────►◄
  └─ -name ── throttle_name ─┘   └─ throttle_name ─┘
```

Parameters

-bandwidth *bandwidth_limit_in_mb*
(Optional) Specifies the bandwidth in MBps. This must be a numeric value 0 - 268435456.

Note: No bandwidth limit is set unless you specify this keyword.

-iops *iops_limit*
(Optional) Specifies the I/O operations limit. This must be a numeric value 0 - 33554432.

Note: No I/O operations limit is set unless you specify this keyword.

-name *throttle_name*
(Optional) Specifies the throttling object's name. This value must be an alphanumeric string up to 63 characters long.

throttle_id | *throttle_name*
(Required) Specifies the volume ID or name of the volume to throttle. The value must be a numeric or alphanumeric string up to 15 characters long.

Description

This command changes attributes associated with a specified throttle object.

An invocation example for changing the bandwidth limit to `100` for an offloaded throttle

```
chthrottle -bandwidth 100 offloadThrottle
```

The detailed resulting output:

```
No feedback
```

An invocation example with no throttling bandwidth limit specified for ID `0`

```
chthrottle -bandwidth 100 0
```

The detailed resulting output:

```
No feedback
```

cleardumps

Use the **cleardumps** command to clear (or delete) the various dump directories on a specified node.

Syntax

```
►►—cleardumps— — -prefix — directory_or_file_filter — ┬────────────┬──────────────────►◄
                                                      ├─ node_id ──┤
                                                      └─ node_name ┘
```

Parameters

-prefix *directory_or_file_filter*
(Required) Specifies the directory, files, or both to be cleared. If a directory is specified, with no file filter, all relevant dump or log files in that directory are cleared. You can use the following directory arguments (filters):

- /dumps (clears all files in all subdirectories)
- /dumps/cimom
- /dumps/cloud
- /dumps/configs
- /dumps/easytier
- /dumps/elogs
- /dumps/feature
- /dumps/iostats

- /dumps/iotrace
- /dumps/mdisk
- /home/admin/update

In addition to the directory, you can specify a filter file. For example, if you specify /dumps/elogs/*.txt, all files in the /dumps/elogs directory that end in .txt are cleared.

Note: The following rules apply to the use of wildcards when using the CLI:

- The wildcard character is an asterisk (*).
- The command can contain a maximum of one wildcard.
- With a wildcard, you must use double quotation marks (" ") around the filter entry, such as in the following entry:

```
>cleardumps -prefix "/dumps/elogs/*.txt"
```

node_id | *node_name*

(Optional) Specifies the node to be cleared. The variable that follows the parameter is either:

- The node name, that is, the label that you assigned when you added the node to the clustered system (system)
- The node ID that is assigned to the node (not the worldwide node name).

Description

This command deletes all the files that match the directory/file_filter argument on the specified node. If no node is specified, the configuration node is cleared.

You can clear all the dumps directories by specifying /dumps as the directory variable.

You can clear all the files in a single directory by specifying one of the directory variables.

You can list the contents of these directories on the given node by using the **lsxxxxdumps** commands.

You can use this command to clear specific files in a given directory by specifying a directory or file name. You can use the wildcard character as part of the file name.

Note: To preserve the configuration and trace files, any files that match the following wildcard patterns are not cleared:

- *svc.config*
- *.trc
- *.trc.old

An invocation example

```
cleardumps -prefix /dumps/configs
```

The resulting output:

```
No feedback
```

An invocation example

```
cleardumps -prefix /dumps/easytier node_2
```

The resulting output:

```
No feedback
```

cpdumps

Use the **cpdumps** command to copy dump files from a nonconfiguration node onto the configuration node.

Note: In the rare event that the /dumps directory on the configuration node is full, the copy action ends when the directory is full and provides no indicator of a failure. Therefore, clear the /dumps directory after migrating data from the configuration node.

Syntax

```
►►— cpdumps — — -prefix ——┬— directory ——┬— —┬— node_name —┬————————————————————————————►◄
                          └— file_filter —┘    └— node_id ———┘
```

Parameters

-prefix *directory* | *file_filter*

(Required) Specifies the directory, or files, or both to be retrieved. If a directory is specified with no file filter, all relevant dump or log files in that directory are retrieved. You can use the following directory arguments (filters):

- /dumps (retrieves all files in all subdirectories)
- /dumps/audit
- /dumps/cimom
- /dumps/cloud
- /dumps/configs
- /dumps/elogs
- /dumps/easytier
- (Storwize V7000)/dumps/enclosure
- /dumps/feature
- /dumps/iostats
- /dumps/iotrace
- /dumps/mdisk
- /home/admin/update

In addition to the directory, you can specify a file filter. For example, if you specified /dumps/elogs/*.txt, all files in the /dumps/elogs directory that end in .txt are copied.

Note: The following rules apply to the use of wildcards with the CLI:

- The wildcard character is an asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, you must surround the filter entry with double quotation marks (""), as follows:

```
>cleardumps -prefix "/dumps/elogs/*.txt"
```

node_id | *node_name*

(Required) Specifies the node from which to retrieve the dumps. The variable that follows the parameter can be one of the following:

- The node name, or label that you assigned when you added the node to the clustered system (system)
- The node ID that is assigned to the node (not the worldwide node name).

If the node specified is the current configuration node, no file is copied.

Description

This command copies any dumps that match the directory or file criteria from the given node to the current configuration node.

You can retrieve dumps that were saved to an old configuration node. During failover processing from the old configuration node to another node, the dumps that were on the old configuration node are not automatically copied. Because access from the CLI is only provided to the configuration node, system files can only be copied from the configuration node. This command enables you to retrieve files and place them on the configuration node so that you can then copy them.

You can view the contents of the directories by using the **lsdumps** command. You can track the status of a copy using the **lscopystatus** command.

An invocation example

```
cpdumps -prefix /dumps/configs nodeone
```

The resulting output:

```
No feedback
```

An invocation example

```
cpdumps -prefix /dumps/easytier node_2
```

The resulting output:

```
No feedback
```

detectiscsistorageportcandidate

Use the **detectiscsistorageportcandidate** command to establish Internet Small Computer Systems Interface (iSCSI) login sessions from any nodes in a specified I/O group to a discovered backend iSCSI target controller.

Syntax

```
►►—detectiscsistorageportcandidate—-srcportid—source_port_id—┬──────────────────────────┬──────────►
                                                              └-iogrp──┬iogrp_id───┬────┘
                                                                       └iogrp_name─┘

►─┬-targetip—ipv4_addr──┬─┬────────────────────────────┬─┬──────────────────────────┬──────────►
  └-targetip6—ipv6_addr─┘ └-username—target_user_name─┘ └-chapsecret—target_chap─┘

►─┬────────────────────────────┬─────────────────────────────────────────────────────────────►◄
  └-site───site_id───site_name─┘
```

Parameters

-srcportid *source_port_id*

(Required) Specifies the source Ethernet port ID (indicated in the **lsportip** output) used to complete target controller discovery. The value must be a number 1 - 8.

- If you also specify **-iogrp**, you trigger discovery through the Ethernet port by using the *source_port_id* specified on all nodes in the I/O group.
- If you do not specify **-iogrp**, you trigger discovery through the Ethernet port by using the *source_port_id* specified on all nodes in the clustered system (system).

-iogrp *iogrp_id* | *iogrp_name*
(Optional) Specifies I/O group ID or name being detected. The *iogrp_id* value must be 0, 1, 2, or 3. The *iogrp_name* value must be an alphanumeric string.

If you specify this parameter, you trigger discovery through the *source_port_id* of both nodes for the specified I/O group. If no value is specified, discovery is triggered by using the *source_port_id* of all nodes in the system.

-targetip *ipv4_addr*
(Required if you do not specify **-targetip6**) Specifies the target iSCSI controller IPv4 address being detected that receives target discovery requests by using the *source_port_id* for all nodes in the specified I/O group.

-targetip6 *ipv6_addr*
(Required if you do not specify **-targetip**) Specifies the target iSCSI controller IPv6 address being detected that receives target discovery requests by using the *source_port_id* for all nodes in the specified I/O group.

-username *target_user_name*
(Optional) Specifies the target controller user name being detected. The value must be an alphanumeric string up to 256 characters.

If the target controller requires a *target_user_name* and *target_chap* for discovery, use the target user name and Challenge-Handshake Authentication Protocol (CHAP) secret to discover the iSCSI target controller.

Some controllers might require that you use the iSCSI qualified name (IQN) user name for discovery. Each nodes IQN is picked up automatically and used if required.

-chapsecret *target_chap*
(Optional) Specifies the *target_chap* required for discovery of the target iSCSI controller that is being detected. The value must be an alphanumeric string (case-sensitive) up to 79 characters.

-site *site_id* | *site_name*
(Optional) Specifies the site ID or site name of the target iSCSI controller that is being detected or discovered. If no I/O group is specified, discovery is done from all nodes present in the specified site and if an I/O group is specified discovery is done only from a node that is part of the specified site and I/O group. The site ID must be 1 (the default) or 2. The site name must be an alphanumeric value.

Important: This parameter must be specified for a HyperSwap or stretched system.

The stretched system topology distributes the I/O group information into each site. Each node in the I/O group is associated with a different site. The back end storage controller of one site is visible only to nodes within that same site. As a result, you attempt storage controller discovery only from nodes that are in the same site.

Description

This command detects iSCSI backend storage controllers for migration and virtualization. This command helps with target iSCSI controller discovery. Use the **lsiscsistorageportcandidate** command to list information about the discovered candidate iSCSI target controller(s).

The target data is available until either another discovery is completed or the system undergoes a recovery procedure, which clears the previous discovery data. The command completes when either discovery from all source nodes completes or the command fails.

Use the **addiscsistorageport** command to establish sessions from any nodes in a specified I/O group to a discovered backend iSCSI controller. Use the **lsiscsistorageportcandidate** command to list information about the candidate iSCSI target controller.

A target discovery invocation example with IP address

This example shows target discovery using an IPv4 IP address with a target ISCSI controller and source port ID 1.

```
detectiscsistorageportcandidate -srcportid 1 -targetip IP1
```

The detailed resulting output:

```
No feedback
```

A target discovery invocation example with source port ID and I/O group

This example shows target discovery from I/O group 3 using an IPv4 address with a target iSCSI controller and source port ID 3.

```
detectiscsistorageportcandidate -srcportid 3 -targetip IP3 -iogrp 3 -username delluser -chapsecret password1
```

The detailed resulting output:

```
No feedback
```

dumpconfig (Discontinued)

Attention: The **dumpconfig** command is discontinued.

help

Use the **help** (or **man**) command to display help information for system commands.

Syntax

```
►►─┬─ help ─┬──┬────────────────┬──────────────────────────────────────►◄
   └─ man ──┘  └─ command_name ─┘
```

Parameters

command_name
  (Optional) Indicates the command name.

Description

Use this command to display help information for system commands. If you specify a command name using *command_name*, the complete help file text for the command is displayed. If you do not specify a command name, a comprehensive list of all commands is displayed (with one brief descriptive line). This list includes these commands:

- satask
- sainfo
- svcconfig
- svc_snap
- svc_livedump

Remember: The **help** command alias is **man**.

An invocation example

```
help
```

The resulting output:

```
addhostiogrp - Maps I/O groups to a host object.
addhostport - Adds worldwide port names (WWPNs) or iSCSI names to a host object.
addmdisk - Adds managed disks to a storage pool.
addnode - Adds a new (candidate) node canister to an existing system.
...
```

lsclustercandidate (Discontinued)

Attention: The **lsclustercandidate** command has been discontinued. Use the **lspartnershipcandidate** command instead.

lscluster (Discontinued)

Attention: The **lscluster** command is discontinued. Use a combination of the **lspartnership**, **lspartnershipcandidate**, and **lssystem** commands instead.

lsclusterip (Discontinued)

Attention: The **lsclusterip** command has been discontinued. Use the **lssystemip** command instead.

lsclusterstats (Discontinued)

Attention: The **lsclusterstats** command is discontinued. Use the **lssystemstats** command instead.

lsdiscoverystatus

Use the **lsdiscoverystatus** command to determine whether a discovery operation is in progress.

Syntax

```
►►— lsdiscoverystatus — ┬──────────────────────────────────┬─ ┬──────────────────┬─ ──────────►
                        └ -filtervalue — attribute_value ─┘  └ -filtervalue? ─┘

►┬─────────┬─ ┬──────────────────────┬──────────────────────────────────────────────────────►◄
 └ -nohdr ─┘  └ -delim — delimiter  ─┘
```

Parameters

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards:

- The wildcard character is the asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, enclose the filter entry within double quotation marks (""):
  `lsdiscoverystatus -filtervalue "IO_group_name=io*"`

-filtervalue?
(Optional) Displays the valid filter attributes for the **-filtervalue** parameter:

- IO_group_id
- IO_group_name
- scope
- status

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

Description

This command displays the state of all discoveries in the cluster. During discovery, the system updates the drive and MDisk records. You must wait until the discovery finishes and is inactive before you attempt to use the system.This command displays one of the following results:

active There is a discovery operation in progress at the time that the command is issued.

inactive
There are no discovery operations in progress at the time that the command is issued.

If the Fibre Channel functions are used only to enable the nodes to cluster, then the Fibre Channel line are not displayed in the **lsdiscoverystatus** command. The *fc_fabric* line appears if there is at least one Fibre Channel controller.

An invocation example

```
lsdiscoverystatus -delim :
```

The resulting output:

```
id:scope:IO_group_id:IO_group_name:status
0:fc_fabric:::active
1:sas_iogrp:0:io_grp0:inactive
3:sas_iogrp:2:io_grp2:active

6:iscsi:::inactive
```

lsfabric

Use the **lsfabric** command to generate a report that displays the Fibre Channel (FC) connectivity between nodes, controllers, and hosts.

Syntax

```
►►── lsfabric ──┬─────────────────────┬── ┬──────────┬── ──────────────────────►
                └─ -delim ── delimiter ┘   └─ -nohdr ─┘
```

```
►─┬──────────────────────────────────────────────────────────┬──┬──────────┬── ─────────────────────►◄
  ├─ -node ──┬─ node_id ───┬─ ─┬──────────────────────┬──────┤  └─ -bytes ─┘
  │          └─ node_name ─┘   └─ -port ── port_id ───┘      │
  ├─ -wwpn ── wwpn ──────────────────────────────────────────┤
  ├─ -host ── host_id_or_name ───────────────────────────────┤
  ├─ -controller ── controller_id_or_name ───────────────────┤
  └─ -cluster ── cluster_id_or_name ─────────────────────────┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (**:**) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-node *node_name* | *node_id*
(Optional) Displays the output for all ports for the specified node. The only parameter that you can specify with the **-node** parameter is the **-port** parameter.

-port *port_id*
(Optional) Displays a concise view of all worldwide port names (WWPNs) that are logged into the specified port ID and node. The **-port** parameter must be specified with only the **-node** parameter. A valid *port_id* value is from a minimum of one through a maximum equal to the number of node Fibre Channel (FC) I/O ports. It specifies the port number in the vital product data (VPD) or the hexadecimal WWPN of the local port.

-wwpn *wwpn*
(Optional) Displays a list of all ports that have a login to the specified WWPN. You cannot use the **-wwpn** parameter with any other parameter.

-host *host_id_or_name*
(Optional) Specifies a host name or ID. Issuing the **lsfabric** command with the **-host** parameter is equivalent to issuing the **lsfabric wwpn** *wwpn* command for every configured WWPN of the specified host. For example, a host with two ports that are zoned to one port of every node in an eight-node clustered system (system) produces 16 lines of output. You cannot use the **-host** parameter with any other parameter.

-controller *controller_id_or_name*
(Optional) Specifies a controller ID or name. You cannot use the **-controller** parameter with any other parameter in this command. Issuing the **lsfabric** command with the **-controller** parameter is equivalent to issuing the **lsfabric wwpn** *wwpn* command for every configured WWPN of the specified controller. For example, a controller with four ports that are connected to an eight node system with two counterpart SANs produces 64 lines of output.

-cluster *cluster_id_or_name*
(Optional) Specifies a system ID or name. You cannot use the **-cluster** parameter with any other parameter. Issuing the **lsfabric** command with the **-cluster** parameter is equivalent to issuing the **lsfabric wwpn** *wwpn* command for every known WWPN in the specified system. Output is sorted by

remote WWPNs and then system WWPNs. This parameter can be used to check the state of connections within the local system or between the local and remote system. When the local system ID or name is specified, each node-to-node connection is listed twice: once from each end. For example, an eight-node system with two counterpart SANs produces eight nodes, which are multiplied by seven other nodes, which are multiplied by two SANs, multiplied by four point-to-point logins, equals 448 lines of output.

Note: The system must be configured in a remote copy partnership with the local system. Remote copy includes Metro Mirror, Global Mirror, and HyperSwap. It must appear in the **`lssystem`** view.

`-bytes`
(Optional) Displays all capacities as bytes.

Description

The **`lsfabric`** command can be issued with any of the parameters to display a limited subset of information. If the command is issued without any parameters, it provides output for every node.

Remember: The value of the `local_port` field is the number of the node's Fibre Channel (FC) port.

Values for the `Type` and `State` columns are:

state active
The meaning of this value depends on the object that it applies to, as follows:

host or controller
Small Computer System Interface (SCSI) commands were issued within the last 5 minutes.

node Node ports can see other ports.

state inactive
No transactions completed within the last 5 minutes.

Note: It can take up to 10 seconds after a command for a controller port to change from inactive to active. It can take up to 5 minutes after a command for a host port to change from inactive to active.

state blocked
This value shows connections that are `blocked` due to the system's port mask settings.

type One of the following values is displayed:

- `host`
- `node`
- `controller`
- `unknown`
- `nas`

You can issue this command to view all the information about the connections that are available to your system.

Remember: The **`lsfabric`** command is limited to displaying 16,384 entries. If you have a large system configuration that exceeds these limits you must filter the output (for example, by node or node port) to view all fabric login records.

An invocation example by using a delimiter

```
lsfabric -delim :
```

The resulting output, in which each row of output contains the following colon-separated columns:

```
remote_wwpn:remote_nportid:id:node_name:local_wwpn:
local_port:local_nportid:state:name:cluster_name:type
```

An invocation example that shows unused (because the system's mask settings are blocked) node logins

```
lsfabric -delim :
```

The resulting output:

```
remote_wwpn:remote_nportid:id:node_name:local_wwpn:local_port:local_nportid:state:name:cluster_name:type
500507680304D190:021700:5:nodeA:500507680304A100:1:020300:active:node4:Cluster_9.115.2:node
500507680304D190:021700:2:nodeB:500507680308A101:2:021800:active:node4:Cluster_9.115.2:node
500507680304D190:021700:3:nodeC:500507680308190D:2:020A00:active:node4:Cluster_9.115.2:node
500507680308D190:011700:5:nodeA:500507680308A100:2:011000:blocked:node4:Cluster_9.115.2:node
500507680308D190:011700:2:nodeB:500507680304A101:1:010D00:blocked:node4:Cluster_9.115.2:node
500507680308D190:011700:3:nodeC:500507680304190D:1:011200:blocked:node4:Cluster_9.115.2:node
```

An invocation example that shows internal connectivity for node 1

```
lsfabric -internal -delim : -node 1
```

The resulting output:

```
remote_wwpn:remote_nportid:id:node_name:local_wwpn:local_port:local_nportid:state:name:cluster_name:type
500507680C520034:010E00:1:node1:500507680C210033:5:010200:active:node2:Cluster_9.19.88:node
500507680C520034:010E00:1:node1:500507680C220033:6:010F00:active:node2:Cluster_9.19.88:node
500507680C520034:010E00:1:node1:500507680C510033:9:010A00:active:node2:Cluster_9.19.88:node
500507680C520034:010E00:1:node1:500507680C520033:10:010B00:active:node2:Cluster_9.19.88:node
500507605EBFEA91:010900:1:node1:500507680C210033:5:010200:active:::expansion
500507605EBFEA91:010900:1:node1:500507680C220033:6:010F00:active:::expansion
500507605EBFEA91:010900:1:node1:500507680C510033:9:010A00:active:::expansion
500507605EBFEA91:010900:1:node1:500507680C520033:10:010B00:active:::expansion
500507605E828601:010100:1:node1:500507680C210033:5:010200:active:::expansion
500507605E828601:010100:1:node1:500507680C220033:6:010F00:active:::expansion
500507605E828601:010100:1:node1:500507680C510033:9:010A00:active:::expansion
500507605E828601:010100:1:node1:500507680C520033:10:010B00:active:::expansion
500507605E828611:010700:1:node1:500507680C210033:5:010200:active:::expansion
500507605E828611:010700:1:node1:500507680C220033:6:010F00:active:::expansion
500507605E828611:010700:1:node1:500507680C510033:9:010A00:active:::expansion
500507605E828611:010700:1:node1:500507680C520033:10:010B00:active:::expansion
500507680C210034:010000:1:node1:500507680C210033:5:010200:active:node2:Cluster_9.19.88:node
500507680C210034:010000:1:node1:500507680C220033:6:010F00:active:node2:Cluster_9.19.88:node
500507680C210034:010000:1:node1:500507680C510033:9:010A00:active:node2:Cluster_9.19.88:node
500507680C210034:010000:1:node1:500507680C520033:10:010B00:active:node2:Cluster_9.19.88:node
500507605EBFEAB1:010400:1:node1:500507680C210033:5:010200:active:::expansion
500507605EBFEAB1:010400:1:node1:500507680C220033:6:010F00:active:::expansion
500507605EBFEAB1:010400:1:node1:500507680C510033:9:010A00:active:::expansion
500507605EBFEAB1:010400:1:node1:500507680C520033:10:010B00:active:::expansion
500507680C510034:010D00:1:node1:500507680C210033:5:010200:active:node2:Cluster_9.19.88:node
500507680C510034:010D00:1:node1:500507680C220033:6:010F00:active:node2:Cluster_9.19.88:node
500507680C510034:010D00:1:node1:500507680C510033:9:010A00:active:node2:Cluster_9.19.88:node
500507680C510034:010D00:1:node1:500507680C520033:10:010B00:active:node2:Cluster_9.19.88:node
500507605EBFEA82:010500:1:node1:500507680C210033:5:010200:active:::expansion
500507605EBFEA82:010500:1:node1:500507680C220033:6:010F00:active:::expansion
500507605EBFEA82:010500:1:node1:500507680C510033:9:010A00:active:::expansion
500507605EBFEA82:010500:1:node1:500507680C520033:10:010B00:active:::expansion
500507605EBFEAA2:010600:1:node1:500507680C210033:5:010200:active:::expansion
500507605EBFEAA2:010600:1:node1:500507680C220033:6:010F00:active:::expansion
500507605EBFEAA2:010600:1:node1:500507680C510033:9:010A00:active:::expansion
500507605EBFEAA2:010600:1:node1:500507680C520033:10:010B00:active:::expansion
500507680C220034:010C00:1:node1:500507680C210033:5:010200:active:node2:Cluster_9.19.88:node
500507680C220034:010C00:1:node1:500507680C220033:6:010F00:active:node2:Cluster_9.19.88:node
500507680C220034:010C00:1:node1:500507680C510033:9:010A00:active:node2:Cluster_9.19.88:node
500507680C220034:010C00:1:node1:500507680C520033:10:010B00:active:node2:Cluster_9.19.88:node
500507605E828631:010800:1:node1:500507680C210033:5:010200:active:::expansion
```

```
500507605E828631:010800:1:node1:500507680C220033:6:010F00:active:::expansion
500507605E828631:010800:1:node1:500507680C510033:9:010A00:active:::expansion
500507605E828631:010800:1:node1:500507680C520033:10:010B00:active:::expansion
500507605E828621:010300:1:node1:500507680C210033:5:010200:active:::expansion
500507605E828621:010300:1:node1:500507680C220033:6:010F00:active:::expansion
500507605E828621:010300:1:node1:500507680C510033:9:010A00:active:::expansion
500507605E828621:010300:1:node1:500507680C520033:10:010B00:active:::expansion
```

lsfcportcandidate

Use the **lsfcportcandidate** command to list the Fibre Channel (FC) ports. This information is used to find open FC ports.

Syntax

```
►►── lsfcportcandidate ── ┬──────────┬── ┬─────────────────────┬──────────────────────►◄
                          └─ -nohdr ─┘   └─ -delim ── delimiter ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (**:**) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

Description

This command returns a list of unconfigured, logged in FC ports.

Note: The **lsfcportcandidate** command presents a list of host FC ports that are logged in to nodes. However, there are situations when the information that is presented might include host FC ports that are no longer logged in or even part of the SAN fabric. For example, if a host FC port is unplugged from a switch but **lsfcportcandidate** shows the worldwide port name (WWPN) that is logged in to all nodes, the incorrect entry is removed when another device is plugged in to the same switch port that previously contained the removed host FC port.

Table 28 shows the possible output:

Table 28. lsfcportcandidate output

Attribute	Description
fc_WWPN	Indicates that the FC WWPN is logged in but unconfigured (not assigned to a host). This value must be 16 hexadecimal characters.

An invocation example

```
lsfcportcandidate
```

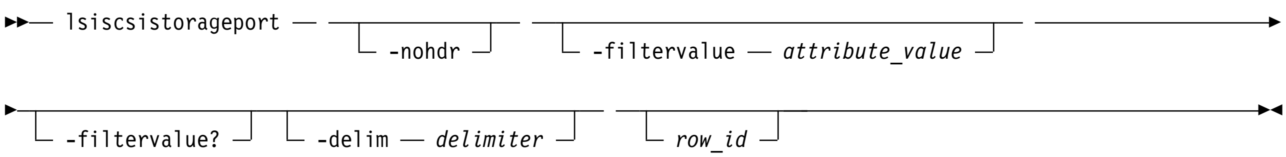
The resulting output:

```
fc_WWPN
200600A0B813B7AC
200600A0B813B7AD
```

lsiscsistorageport

Use the **lsiscsistorageport** command to display the details of the iSCSI login sessionsthat are established from the Initiator's Internet Small Computer Systems Interface (iSCSI) source ports to iSCSI backend target controller ports.

Syntax



Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards:

- The wildcard character is the asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, enclose the filter entry within double quotation marks (""):
  `lsiscsistorageport -filtervalue id="1*"`

-filtervalue?
(Optional) Displays the valid filter attributes for the **-filtervalue** parameter:

- id
- status

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (**:**) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

row_id
(Optional) Specifies the row ID view and denotes the sessions that are established from the specified initiator node ports to a backend controller target iSCSI qualified name (IQN) through a target Internet Protocol (IP) address. The value must be a number 0 - 1024.

Description

This command displays the details of sessions that are established from the Initiator's iSCSI source ports to iSCSI backend target controller ports.

This table provides the attribute values that can be displayed as output view data.

Table 29. lsiscsistorageport output

Attribute	Description
id	Indicates the object identifier for any sessions from any clustered system (system) nodes to the iSCSI backend controller iSCSI qualified name (IQN) through an iSCSI backend controller target IP. The value must be a number 0 - 1023.
src_port_id	Indicates the source port identifier for the node Ethernet port number that is displayed in the **lsportip** output. The value is a number 0 - 7.
target_ipv4	Indicates the IPv4 address of the iSCSI backend controller target port that establishes a session from the initiator source port that is identified by the source port ID. The default value is blank.
target_ipv6	Indicates the IPv6 address of the iSCSI backend controller target port that establishes a session from the initiator source port that is identified by the source port ID. The default value is blank.
target_iscsiname	Indicates the IQN of the iSCSI backend controller target that establishes a session. The value must be an alphanumeric string of no more than 256 characters. The default value is blank.
controller_id	Indicates the controller ID that is displayed in the **lscontroller** output. The value must be a number 0 - 1023. The default value is 1024.
iogroup_list	Indicates a colon-separated list of discovery result codes: The value must be an alphanumeric string of up to 32 characters. This field cannot be blank. The values are `0` and `1`: • `0` indicates that the I/O group is available in the system, but discovery is either not triggered through the I/O group or discovery through the I/O group fails. • `1` indicates that the I/O group is present and discovery is successful through the I/O group. **Note:** The value `-` (dash) indicates that the I/O group is not valid or is not present in the system.
status	Indicates the connectivity status from all nodes in the system to the target port. The values are: • `full` If you specify a single I/O group by using the **addiscsistorageport** command and you establish the session from all nodes in the specified I/O group, the status is `full`. • `partial` If you specify a single I/O group by using the **addiscsistorageport** command and you establish the session from a single node in the specified I/O group, the status is `partial`. • `none` If you specify a single I/O group by using the **addiscsistorageport** command and you do not establish the session from any node in the specified I/O group, the status is `none`. There is no default value. This field cannot be blank.

Table 29. lsiscsistorageport output (continued)

Attribute	Description
connected	Indicates whether the established connection is from a specified Ethernet port of a target IQN and IP address. The values are `yes` and `no`.
site_id	Indicates the site ID (if the nodes that are being discovered belong to a site). This applies to stretched and HyperSwap systems.
site_name	Indicates the site name (if the nodes that are being discovered belong to a site). This applies to stretched and HyperSwap systems.
node_id	Indicates the node ID of the initiator node that establishes the session. The value must be a numeric value.
node_name	Indicates the node name of the initiator node that establishes the session. The value must be an alphanumeric string of no more than 16 characters (the default value is blank).
src_ipv4	Indicates the IPv4 IP address of the source port ID on a specified node. The default value is blank.
src_ipv6	Indicates the IPv6 IP address of the source port ID on a specified node. The default value is blank.
src_iscsiname	Indicates the IQN of the source node for which connectivity is being displayed for the target port. The value must be an alphanumeric string of no more than 256 characters (the default value is blank).

Before you specify **lsiscsistorageport** such as in the examples below, you must:

1. Complete target discovery by using an IPv4 IP address of a target ISCSI controller through source port ID `0`:

   ```
   detectiscsistorageportcandidate -targetip IP1 –srcportid 2
   ```

2. You would then specify session establishment by using **addiscsistorageport** for discovery output row `0` through I/O group 1:

   ```
   addiscsistorageport -iogrp 1 0
   ```

3. Specify **lsiscsistorageport** to view the output (no *tgt_user_name* or *target_chap* is required for discovery or session establishment).

Specify **rmiscsistorageport** to remove a session.

A concise invocation example

```
lsiscsistorageport
```

The resulting output:

```
id src_port_id target_ipv4   target_ipv6   target_iscsiname                    controller_id  controller_name  iogroup_li
0  4           192.168.82.90               iqn.1986-03.com.ibm:2145.temp.node1  3              controller3      0:1:-:-
```

A detailed invocation example

```
lsiscsistorageport 0
```

The resulting output:

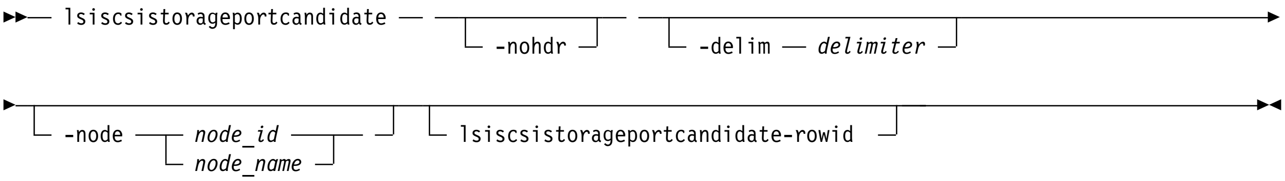
```
id 0
src_port_id 4
target_ipv4 192.168.82.90
target_ipv6
target_iscsiname iqn.1986-03.com.ibm:2145.temp.node1
controller_id 0
iogroup_list 1:1:-:-
status full
site_id
```

```
site_name
node_id 17
node_name node1
src_ipv4 192.168.82.80
src_ipv6
src_iscsiname iqn.1986-03.com.ibm:2145.iscsicluster.node1
connected yes
node_id 20
node_name node2
src_ipv4 192.168.82.81
src_ipv6
src_iscsiname iqn.1986-03.com.ibm:2145.iscsicluster.node2
connected yes
node_id 16
node_name node3
src_ipv4 192.168.82.82
src_ipv6
src_iscsiname iqn.1986-03.com.ibm:2145.iscsicluster.node3
connected yes
node_id 18
node_name node4
src_ipv4 192.168.82.83
src_ipv6
src_iscsiname iqn.1986-03.com.ibm:2145.iscsicluster.node4
connected yes
```

lsiscsistorageportcandidate

Use the **lsiscsistorageportcandidate** command to display a concise or detailed list of information about the candidate Internet Small Computer Systems Interface (iSCSI) target controller iSCSI Qualified Name (IQN) that is specified with the target IP from the specified initiator source ports.

Syntax


lsiscsistorageportcandidate
-nohdr
-delim
delimiter
-node
node_id
node_name
lsiscsistorageportcandidate-rowid

Parameters

-nohdr

(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If no data is available to be displayed, headings are not displayed.

-node *node_name* | *node_id*

(Optional)

Specifies the ID or name of a node in the system. The value must be an alphanumeric string.

-delim *delimiter*

(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

`lsiscsistorageportcandidate-rowid`
(Optional) Specifies a row ID shown in the concise view to provide a detailed view of information about discovered portal IPs.

Description

This command lists information about the last invocation of the **`detectiscsistorageportcandidate`** command. This command can also display two types of lists:

- A concise list of information about the candidate iSCSI target controller IQNs that are visible at the specified target IP from the specified initiator ports along with indication of which initiator ports can see each discovered iSCSI target IQN.
- The detailed information about all the Target Controller Portal IPs discovered during the last invocation of the **`detectiscsistorageportcandidate`** command.

This table provides the attribute values that can be displayed as output view data.

Table 30. lsiscsistorageportcandidate output

Attribute	**Description**
id	Indicates the row ID for the discovery output. Enter the **`detectiscsistorageportcandidate`** command before you use the **`lsiscsistorageportcandidate`** command. Display the concise view first to show one row per IQN. Use a row ID from the concise view to specify the detailed view of the **`lsiscsistorageportcandidate`** *lsiscsistorageportcandidate-rowid* command. The detailed view displays the list of discovered target IP addresses for the IQN.
src_port_id	Indicates the source port ID. The value is a number 1 - 8.
target_ipv4	Indicates the target IPv4 address.
target_ipv6	Indicates the target IPv6 address.
target_iscsiname	Indicates the (discovered) IQN that uses the target controller. The value is an alphanumeric string that is 256 characters long.
iogroup_list	Indicates a colon-separated list of discovery result codes: The value must be an alphanumeric string of up to 32 characters. This field cannot be blank. The values are `0` and `1`: • `0` indicates that the I/O group is available in the system, but discovery is either not triggered through the I/O group or discovery through the I/O group fails. • `1` indicates that the I/O group is present and discovery is successful through the I/O group. **Note:** The value `-` (dash) indicates that the I/O group is not valid or is not present in the system.
status	Indicates whether discovery was successful. The status is one of the following values: • `full` • `partial` • `none`
configured	Indicates whether the discovered target IQN has any established sessions with source ports or target ports. The values are `yes` and `no` (default).
site_id	Indicates the site ID (if the nodes that are being discovered belong to a site). This attribute applies to stretched and HyperSwap systems.
site_name	Indicates the site name (if the nodes that are being discovered belong to a site). This attribute applies to stretched and HyperSwap systems.

A concise invocation example

First, you must specify target discovery by using an IPv4 IP address for a target ISCSI controller through the source port ID 1. No *tgt_user_name* or *target_chap* value is required.

```
detectiscsistorageportcandidate –srcportid 1 -targetip 10.10.10.1
```

To view the output, specify the following command:

```
lsiscsistorageportcandidate
```

The following concise resulting output is displayed:

```
id  src_port_id target_ipv4 target_ipv6 target_iscsiname  iogroup_list configured status site_id site_name
0   1           10.10.10.1              IQN1              1:1:1:1      no         Full
```

A concise invocation example

This example shows target discovery by using an IPv4 IP address for a target ISCSI controller through I/O group `0` and source port ID `0`:

```
detectiscsistorageportcandidate -iogrp 0 –srcportid 1 -targetip IP2  –username superuser –chapsecret password2
```

A *tgt_user_name* and *target_chap* value are used. This system has only two I/O groups, `0` and `3`.

To view the output, specify the following command:

```
lsiscsistorageportcandidate
```

The following concise resulting output is displayed:

```
id  src_port_id target_ipv4 target_ipv6 target_iscsiname  iogroup_list configured status site_id site_name
0   1           IP2                     IQN1              1:-:-:0      no         Full
1   1           IP2                     IQN2              1:-:-:0      no         Full
2   1           IP2                     IQN3              1:-:-:0      no         Full
```

A detailed invocation example

First, you must specify target discovery by using an IPv4 IP address for a target ISCSI controller through the source port ID 1 and a target IP address.

```
detectiscsistorageportcandidate –srcportid 1 -targetip 10.10.10.1
```

The concise view has one row per IQN as shown in the first concise view example. Each row ID from the concise view identifies an iSCSI qualified name (IQN). To view detailed information for the IQN, specify the row ID (id):

```
lsiscsistorageportcandidate 0
```

The resulting output lists details of the discovered portal IP addresses for the IQN:

```
SendTargets Portal IPs
10.10.10.1
10.10.10.2
fe:65::00:01
fe:65::00:02
```

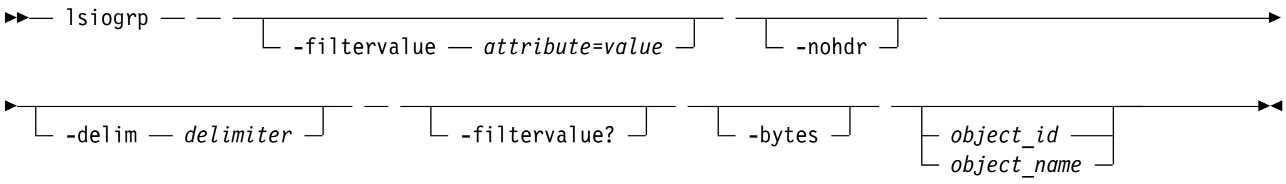
lsiogrp

Use the **`lsiogrp`** command to display a concise list or a detailed view of input/output (I/O) groups visible to the system.

The list report style can be used to obtain the following two styles of report:

- A list that contains concise information about all the I/O groups that are visible to the system. Each entry in the list corresponds to a single I/O group.
- The detailed information about a single I/O group.

Syntax



Parameters

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are returned. If a capacity is specified, the units must also be included.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcard characters when you use the CLI:

- The wildcard character is an asterisk (*), which must be the first or last character in the string.
- The command can contain a maximum of one wildcard.
- When you use a wildcard, enclose the filter entry within double quotation marks (""), as follows:
  - `lsiogrp -filtervalue "name=md*"`

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-filtervalue?
(Optional) Displays the valid filter attributes for the **lsiogrp** command:

- `HWS_name`
- `HWS_unique_id`
- `node_count`
- `name`
- `id`
- `host_count`

-bytes
(Optional) Displays all capacities as bytes.

object_id | *object_name*
(Optional) Specifies the name or ID of an object. When you use this parameter, the detailed view of the specific object is returned and any value that is specified by the **-filtervalue** parameter is

ignored. If you do not specify the *object_id* | *object_name* parameter, the concise view of all objects that match the filtering requirements that are specified by the **-filtervalue** parameter are displayed.

Description

This command returns a concise list or a detailed view of I/O groups visible to the system.

You can display the following information for this command:

id Indicates the ID of the I/O group.

name Indicates the name of the I/O group.

node_count
Indicates the number of nodes in the I/O group.

vdisk_count
Indicates the number of volumes in the I/O group.

host_count
Indicates the number of hosts in the I/O group.

flash_copy_total_memory
Indicates the total amount of memory that is allocated to FlashCopy.

flash_copy_free_memory
Indicates the total amount of memory that is allocated to FlashCopy, but unused.

remote_copy_total_memory
Indicates the total amount of memory that is allocated to Remote Copy, but unused. Remote copy includes Metro Mirror, Global Mirror, and HyperSwap.

remote_copy_free_memory
Indicates the total amount of memory that is allocated to Remote Copy, but unused. Remote copy includes Metro Mirror, Global Mirror, and HyperSwap.

mirroring_total_memory
Indicates the total amount of memory that is allocated to mirroring.

raid_total_memory
Indicates the total amount of memory that is allocated to RAID.

raid_free_memory
Indicates the total amount of memory that is allocated to RAID, but is unused.

maintenance
Indicates whether the I/O group is in maintenance mode. The values are:

- `yes`
- `no`

compression_active
Indicates whether Real-time compression is used in the selected I/O group.

accessible_vdisk_count
The number of accessible volumes in this I/O group.

compression_supported
Indicates whether the I/O group supports Real-time compression or data reduction compression.

max_enclosures
Indicates the maximum number of enclosures that are supported by this I/O group.

encryption_supported
Indicates whether the I/O group supports encryption for attached drives. The possible values are:

- yes
- no

flash_copy_maximum_memory
Identifies the maximum memory that can be set for the specified I/O group. The value must be a numeric string in the range 552 MB - 2048 MB.

site_id
Identifies the site ID for the I/O group. The possible values are:

- 1
- 2

site_name
Identifies the site name for the I/O group. The value must be an alphanumeric string or blank.

compression_total_memory
Indicates the total amount of memory that is allocated for Real-time compression per node in the specified I/O group.

fctargetportmode
Indicates the current N_Port ID Virtualization (NPIV) status in the specified I/O group. The values are:

- disabled
- transitional
- enabled
- changing_disabled_to_transitional
- changing_transitional_to_disabled
- changing_enabled_to_transitional
- changing_transitional_to_enabled

1 **deduplication_supported**
1 Indicates whether this I/O group supports data deduplication. The value that is displayed is
1 either yes or no.

1 **Note:** An I/O group indicates that data deduplication is supported if the nodes in the I/O group
1 have 32 GB of memory (or greater) installed. The existence of Real-time compression volumes in
1 the I/O group does not influence whether data deduplication is shown as supported.

1 **deduplication_active**
1 Indicates whether data deduplication is in use in the I/O group. The value that is displayed is
1 either yes or no.

A concise invocation example

```
lsiogrp -delim :
```

The resulting output:

```
id:name:node_count:vdisk_count:host_count:site_id:site_name
0:io_grp0:1:0:0:1:chelsea1
1:io_grp1:0:0:0:2:chelsea2
2:io_grp2:0:0:0:3:chelsea1
3:io_grp3:0:0:0:4:chelsea1
4:recovery_io_grp:0:0:0:5:chelsea1
```

A detailed invocation example

```
lsiogrp -delim : 0
```

The detailed output:

```
id:0
name:io_grp0
node_count:1
vdisk_count:51
host_count:0
flash_copy_total_memory:3.0MB
flash_copy_free_memory:1.0MB
remote_copy_total_memory:6.5MB
remote_copy_free_memory:2.8MB
mirroring_total_memory:1.0MB
mirroring_free_memory:0.3MB
raid_total_memory:2MB
raid_free_memory:25.0MB
maintenance: no
compression_active:yes
accessible_vdisk_count:10
compression_supported:yes
max_enclosures:21
encryption_supported:yes
flash_copy_maximum_memory:2048.0MB
site_id:2
site_name:chelsea2
compression_total_memory:35128.0MB
fctargetportnode:disabled
deduplication_supported: yes
deduplication_active: no
```

lshbaportcandidate (Deprecated)

The **lshbaportcandidate** command is deprecated. Use either the **lsfcportcandidate** or **lssasportcandidate** command instead.

lsiogrphost

Use the **lsiogrphost** command to display a list of the hosts mapped to a specified I/O group.

Syntax

```
►►── lsiogrphost ── ─┬──────────┬─ ─┬──────────────────────┬─ ┬─ iogrp_id ───┬──────────────►◄
                     └─ -nohdr ─┘   └─ -delim ── delimiter ─┘  └─ iogrp_name ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim ***delimiter***
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim : on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

iogrp_id | *iogrp_name*
(Required) The ID or name of the I/O group for which a list of all mapped hosts is required.

Description

The **lsiogrphost** command displays a list of hosts that are mapped to a specified I/O group.

An invocation example

```
lsiogrphost -delim : 0
```

The resulting output:

```
id:name
0:hostzero
1:hostone
```

lsiogrpcandidate

Use the **lsiogrpcandidate** command to list the I/O groups that can have nodes added to them.

Syntax

```
►►— lsiogrpcandidate — ┬─────────┬─ ┬─────────────────────┬─────────────────────►◄
                       └─ -nohdr ┘  └─ -delim — delimiter ┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim : on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

Description

The **lsiogroupcandidate** command displays a list of I/O groups to which nodes can be added. Only the I/O group IDs are displayed.

An invocation example

```
lsiogrpcandidate
```

The resulting output:

```
id
0
1
2
3
4
```

lsiostatsdumps (Deprecated)

Attention: The **lsiostatsdumps** command is deprecated. Use the **lsdumps** command to display a list of files in a particular dumps directory.

lsiotracedumps (Deprecated)

Attention: The **lsiotracedumps** command is deprecated. Use the **lsdumps** command to display a list of files in a particular dumps directory.

lsnode (SVC) / lsnodecanister (Storwize family products)

Use the **lsnode**/ **lsnodecanister** command to return a concise list or a detailed view of nodes or node canisters that are part of the clustered system (system).

The list report style can be used to obtain two styles of report:

- A list containing concise information about all the nodes or node canister on a system. Each entry in the list corresponds to a single node or node canister.
- The detailed information about a single node or node canister.

Syntax

```
►►─ lsnode ─ | lsnodecanister ─ ─┬──────────────────────────────────┬─ ─┬───────────┬─ ──────────►
                                 └─ -filtervalue ─ attribute=value ─┘   └─ -nohdr ─┘

►─┬──────────────────────┬─ ─ ─┬────────────────┬─┬───────────────┬─────────────────────────►◄
  └─ -delim ─ delimiter ─┘     └─ -filtervalue? ─┘ ├─ object_id ───┤
                                                  └─ object_name ─┘
```

Parameters

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are returned. If a capacity is specified, the units must also be included.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards with the Command-Line Interface (CLI):

- The wildcard character is an asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, you must enclose the filter entry within double quotation marks (""):

```
lsnode -filtervalue "name=md*"
```

-filtervalue?
Displays a list of valid filter attributes for the **-filtervalue***attribute=value* parameter. The valid filters for the **lsnode** command are:

- canister_id
- config_node/config_nodecanister
- enclosure_id
- enclosure_serial_number
- hardware
- id
- iscsi_alias
- IO_group_name

- IO_group_id
- name
- panel_name
- status
- site_id

-nohdr

(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*

(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

object_id | *object_name*

(Optional) Specifies the object ID or name. When you use this parameter, the detailed view of the specific object is returned and any value that is specified by the **-filtervalue** parameter is ignored. If you do not specify the *object_id* | *object_name* parameter, the concise view of all objects that match the filtering requirements that are specified by the **-filtervalue** parameter are displayed.

Description

This command returns a concise list or a detailed view of nodes or node canisters that are part of the system. Table 31 provides the possible values that are applicable to the attributes that are displayed as data in the output views.

Table 31. lsnode or lsnodecanister attribute values

Attribute	Value
status	Indicates the status. The values are: • `offline` • `service` • `flushing` • `pending` • `online` • `adding` • `deleting` • `spare` • `online_spare`
config_node	Indicates if the node is a configuration node. The values are: • `yes` • `no`
IO_group_id	Indicates the I/O group of the node. **Note:** A node that is considered a spare node does not use an I/O group ID for a node (spare or offline) that has been replaced by an online spare node.

Table 31. lsnode or lsnodecanister attribute values (continued)

Attribute	Value
IO_group_name	Indicates the name of the I/O group in the node. **Note:** A node that is considered a spare node does not use an I/O group name for a node (spare or offline) that has been replaced by an online spare node.
port_status	Indicates whether the node is a configuration node. The values are: • `active` • `inactive` • `not_installed`
hardware	Indicates the hardware type (for example, `DH8`).
UPS_serial_number	Indicates the serial number of the UPS.
UPS_unique_id	Indicates the unique ID of the UPS.
panel_name	Indicates the unique identifier for the node.
enclosure_id	Blank. This field is blank for a node-based system.
canister_id	Blank. This field is blank for a node-based system.
enclosure_serial_number	Blank. This field is blank for a node-based system.
service_IP_mode	Indicates the current mode of the service IPv4 • Empty if IPv4 is not active • The values are: – `static` (if the service IP is set by the user) – `dhcp` (if the service IP is set successfully by using DHCP server) – `dhcpfallback` (if the service IP is set to a default value after a DHCP server request failed)
service_IP_mode_6	Indicates the current mode of the service IPv6 • Empty if IPv6 is not active • Either static (if the service IP is set by the user) or dhcp (if the service IP set successfully by using DHCP server).
site_id	Indicates the site node value.
site_name	Indicates the site name.
identify_LED	Indicates the node or node canister state - `on`, `off`, or blank.
product_mtm	Indicates the product machine type.
code_level	Indicates the current level of machine code that is running on the node. `on`, `off`, or blank.
serial_number	Indicates the current serial number for the node.
machine_signature	Indicates the current machine signature for the node.
update_complete	Indicates whether the node completes its update. The value is `yes` or `no`.
spare	Indicates whether the node is a spare. The value is `yes` or `no`.
failover_source	Indicates the node ID for a node that fails over to a specified node. This value is always blank if the node status is not `online_spare`.
protected_nodes	Indicates the nodes that might fail over to a specified node. The value is blank unless the node status is `spare`. **Remember:** This value does not indicate whether there is source node redundancy.

The first four Fibre Channel (FC) input/output (I/O) ports display the worldwide port name (WWPN), state, and speed. If there are less than four FC I/O ports, the fields display with a WWPN of 0000000000000000, port_status of inactive, and port_speed of N/A. To examine the FC ports, use the **lsportfc** command.

A node in the spare state displays a blank value for:

- IO_group_id
- IO_group_name
- partner_node_id
- partner_node_name.

A node with an online_spare state has a valid IO_group_id and IO_group_name, and might also have a valid partner_node_id and partner_node_name. The IO_group_id and IO_group_name values are blank when the node is in spare state.

Remember: Nodes that are in spare state or online_spare state must have a valid and unique node ID.

A concise invocation example for SAN Volume Controller

```
lsnode
```

The concise resulting output:

```
id name  UPS_serial_number WWNN             status IO_group_id IO_group_name config_node UPS_unique_id hardware iscsi_name
1  node1                   500507680C000128 online 0           io_grp0       yes                       SV1      iqn.1986-03
2  node2                   500507680C000130 online 0           io_grp0       no                        SV1      iqn.1986-03
3  node3                   500507680C000138 online 1           io_grp1       no                        SV1      iqn.1986-03
4  node4                   500507680C000140 online 1           io_grp1       no                        SV1      iqn.1986-03
5  node5                   500507680C000148 online 2           io_grp2       no                        SV1      iqn.1986-03
6  node6                   500507680C000180 online 2           io_grp2       no                        SV1      iqn.1986-03
7  node7                   500507680100A283 online 3           io_grp3       no                        SV1      iqn.1986-03
8  node8                   500507680100A284 online 3           io_grp3       no                        SV1      iqn.1986-03
```

A concise invocation example for SAN Volume Controller

```
lsnode -delim ,
```

The concise resulting output:

```
id,name,UPS_serial_number,WWNN,status,IO_group_id,IO_group_name,config_node,UPS_unique_id,hardware,iscsi_name,iscsi_alias,p
1,node114120,UPS_Fake_SN,5005076801005D00,online,0,io_grp0,yes,1000000000005D00,DH8,iqn.1986-03.com.ibm:2145.mycluster.node
```

A concise invocation example for Storwize Family products

```
lsnodecanister -delim ,
```

The concise resulting output:

```
id,name,UPS_serial_number,WWNN,status,IO_group_id,IO_group_name,config_node,UPS_unique_id,hardware,iscsi_name,iscsi_alias,p
1,node114120,UPS_Fake_SN,5005076801005D00,online,0,io_grp0,yes,1000000000005D00,DH8,iqn.1986-03.com.ibm:2145.mycluster.node
```

A detailed invocation examplefor SAN Volume Controller

```
lsnode -delim , 1
```

The resulting output:

```
id,1
name,hlcn114289
UPS_serial_number,10004BC018
WWNN,5005076801002978
status,online
IO_group_id,0
IO_group_name,io_grp0
```

```
partner_node_id,2
partner_node_name,hlcn114253
config_node,no
UPS_unique_id,20400001124C0048
port_id,5005076801402978
port_status,active
port_speed,4Gb
port_id,5005076801302978
port_status,active
port_speed,4Gb
port_id,5005076801102978
port_status,active
port_speed,4Gb
port_id,5005076801202978
port_status,active
port_speed,4Gb


hardware,DH8
iscsi_name,iqn.1986-03.com.ibm:2145.ldcluster-19.hlcn114289
iscsi_alias,
failover_active,no
failover_name,hlcn114253
failover_iscsi_name,iqn.1986-03.com.ibm:2145.ldcluster-19.hlcn114253
failover_iscsi_alias,
panel_name,114289
enclosure_id,
canister_id,
enclosure_serial_number,
service_IP_address,9.180.29.52
service_gateway,9.180.28.1
service_subnet_mask,255.255.254.0
service_IP_address_6,
service_gateway_6,
service_prefix_6,
service_IP_mode,dhcp
service_IP_mode_6
site_id,1
site_name,DataCenterA

identify_LED,on
product_mtm,2145-DH8
code_level,7.4.0.0 (build 99.1.1406102000)
serial_number,78G0123
machine_signature,0123-4567-89AB-CDEF
spare,yes
failover_source
protected_nodes 1,2
```

A detailed invocation example for Storwize Family products

```
lsnodecanister -delim , 1
```

The resulting output:

```
id,1
name,hlcn114289
UPS_serial_number,10004BC018
WWNN,5005076801002978
status,online
IO_group_id,0
IO_group_name,io_grp0
partner_nodecanister_id,2
partner_nodecanister_name,hlcn114253
config_nodecanister,no
UPS_unique_id,20400001124C0048
port_id,5005076801402978
port_status,active
```

```
port_speed,4Gb
port_id,5005076801302978
port_status,active
port_speed,4Gb
port_id,5005076801102978
port_status,active
port_speed,4Gb
port_id,5005076801202978
port_status,active
port_speed,4Gb
hardware,DH8
iscsi_name,iqn.1986-03.com.ibm:2145.ldcluster-19.hlcn114289
iscsi_alias,
failover_active,no
failover_name,hlcn114253
failover_iscsi_name,iqn.1986-03.com.ibm:2145.ldcluster-19.hlcn114253
failover_iscsi_alias,
panel_name,114289
enclosure_id,
canister_id,
enclosure_serial_number,
service_IP_address,9.180.29.52
service_gateway,9.180.28.1
service_subnet_mask,255.255.254.0
service_IP_address_6,
service_gateway_6,
service_prefix_6,
service_IP_mode,dhcp
service_IP_mode,
identify_LED,on
site_id,1
site_name,DataCenterA
identify_LED
product_mtm 2145-DH8
code_level,7.4.0.0 (build 99.1.1406102000)
serial_number 78G0123
machine_signature 0123-4567-89AB-CDEF
spare,yes
failover_source
protected_nodes 1,2
```

lsnodebattery

Use the **lsnodebattery** command to display information about the batteries in a node. This command applies to SAN Volume Controller 2145-DH8 systems.

Syntax

```
►►—lsnodebattery— ┬──────────────────────┬ ┬─────────┬ ┬───────────────────────┬ ──────────►
                  └ -delim — delimiter ┘  └ -nohdr ┘  └-battery—battery_id┘

►─┬──────┬───────────────────────────────────────────────────────────────────────────►◄
  └node┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-battery *battery_id*
(Optional) Specifies the battery ID. If you specify this parameter, you must also specify *node*.

node
(Optional) Specifies the node ID or name.

Description

The command displays information about the batteries in a node. The concise view displays a line for each battery slot in all nodes.

Table 32 provides the possible values that are applicable to the attributes that are displayed as data in the output views.

Table 32. lsnodebattery attribute values

Attribute	**Value**
node_id	Identifies the node that contains the battery.
node_name	Identifies the node that contains the battery.
battery_id	Identifies the battery in the node.
status	Identifies the status of the battery: • `online` if the battery is present and working as usual (which includes a battery calibration). • `degraded` indicates that the battery is present but not working as usual (it has an error logged against it). • `offline` indicates that the battery cannot be detected or is failed (a node error indicating it is missing or failed is logged against the battery). **Remember:** If the status is `offline`, all other fields display the most recent battery data. If no data was shown, all fields remain blank.
charging_status	Identifies the charging state of the battery: • `charged` indicates that the battery is fully charged. • `charging` indicates that the battery is charging. • `discharging` indicates that the battery is losing voltage (life) and it is recalibrating the gas gauge after the battery recharges. • `idle` indicates that the battery is not charging and it is not discharging, but it is not fully charged. • `reconditioning` indicates that the battery is reconditioning itself by discharging and after recharging. **Important:** Reconditioning occurs approximately every three months (on redundant systems) and can take from 12 - 48 hours.
recondition_needed	Identifies that the battery needs to be reconditioned or it must start reconditioning soon. A recalibration is required because the reported values from the gas gauge are not trusted. The values are `yes` and `no`.
node_percentage_charge	Identifies the battery's percentage charged to determine if it can support the node.

Table 32. lsnodebattery attribute values (continued)

Attribute	Value
end_of_life_warning	Identifies the battery's end of life (with a warning noise). The values are `yes` and `no`. **Important:** Replace the battery.
present	Identifies a battery is present. The values are `yes` and `no`.
redundant	Identifies if it is safe to remove the battery. The values are `yes` and `no`.
remaining_charge_capacity_mAh	Identifies the remaining capacity of the battery (when the battery recharges) in milliamps hours (mAh).
full_charge_capacity_mAh	Identifies the fully charged capacity of the battery in mAh.
FRU_part_number	Identifies the FRU part number of the battery. The value contains seven alphanumeric characters.
FRU_identity	Identifies the 11S number (combining the manufacturing part number and the serial number). The value contains 22 alphanumeric characters.
compatibility_level	Identifies the battery driver software must support the current software level to operate with this battery (this comes from the battery vital product data or VPD).
last_recondition_timestamp	Indicates a system timestamp for the last successful recalibration of the gas gauge. The format is `YYMMDDHHMMSS`, where: • `YY` indicates year. • The first `MM` indicates month. • `DD` indicates day. • `HH` indicates hour. • The second `MM` indicates minute. • `SS` indicates second.
powered_on_hours	Indicates the number of hours the battery is in a powered node.
cycle_count	Identifies the number of charge or discharge cycles the battery performs.
error_sequence_number	Identifies the error log number of the highest priority error. This field is generally blank, but if the status is `degraded` or `offline` an error sequence number is displayed.

A concise invocation example

If battery `2` in node `1` is not installed:

```
lsnodebattery
```

The resulting output:

```
node_id node_name battery_id status  charging_status recondition_needed  node_percentage_charge   end_of_life_warning
1       node1     1          online  charged         no                  50                       no
1       node1     2          offline
2       node2     1          online  charged         no                  50                       no
2       node2     2          online  reconditioning  yes                 50
```

A concise invocation example

If battery `2` in node `1` is failing to charge:

```
lsnodebattery
```

The resulting output:

```
node_id node_name battery_id status   charging_status recondition_needed  node_percentage_charge   end_of_life_warning
1       node1     1          online   charged         no                  50                       no
1       node1     2          offline  idle            no                  50                       no
2       node2     1          online   charged         no                  100                      no
2       node2     2          online   charged         no                  100
```

A concise invocation example

If battery 2 in node 1 is removed (last known status is `presented`):

```
lsnodebattery
```

The resulting output:

```
node_id node_name battery_id status  charging_status recondition_needed  node_percentage_charge   end_of_life_warning
1       node1      1          online  charged         no                  50                       no
1       node1      2          offline charged         no                  50                       no
2       node2      1          online  charged         no                  50                       no
2       node2      2          online  reconditioning  yes                 50
```

A detailed invocation example

```
lsnodebattery -battery 2 2
```

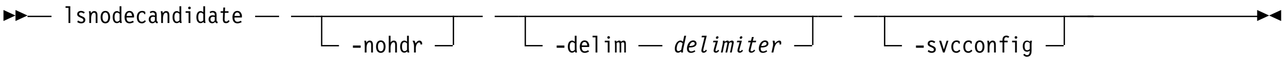
The resulting output:

```
node_id 2
node_name node2
battery_id 2
status online
charging_status reconditioning
present yes
redundant yes
recondition_needed yes
remaining_charge_capacity_mAh 1600
full_charge_capacity_mAh 1950
end_of_life_warning no
FRU_part_number FRU0001
FRU_identity 11SYM30BG123456MAN0001
compatability_level 5
last_recondition_timestamp 0
powered_on_hours 12345
cycle_count 2
node_percentage_charge 50
error_sequence_number
```

lsnodecandidate (SAN Volume Controller)

Use the **lsnodecandidate** command to list all of the nodes that are available to add to the clustered system.

Syntax



Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-svcconfig
(Optional) Lists all nodes in the enclosure that are in a candidate state.

Description

Note: The **lsnodecandidate** command is a SAN Volume Controller command. For Storwize V7000, use the **lscontrolenclosurecandidate** command.

This command displays a list of nodes that are available to add to the clustered system. This list includes nodes that are not already part of a clustered system, but are compatible with the clustered system code level. Nodes with hardware types that are incompatible with the installed code are not listed.

The following table describes the possible outputs:

Table 33. lsnodecandidate outputs

Attribute	**Description**
panel_name	Unique identifier for the node.
UPS_serial_number	The serial number of the UPS.
UPS_unique_id	The unique ID of UPS.
hardware	Describes the type of nodes.
serial_number	Indicates the current serial number for the node (7 characters).
product_mtm	Indicates the current product machine type for the node (8 characters with the hyphen).
machine_signature	Indicates the current machine signature for the node (16-character hexadecimal string with hyphens).

An invocation example

```
lsnodecandidate -delim :
```

The resulting output:

```
id: panel_name:UPS_serial_number:UPS_unique_id:hardware:serial_number:product_mtm:machine_signature
1:146355:10L3ASH:202381001C0D18D8:8G4:78G0123:2145-DH8:0123-4567-89AB-CDEF
```

An invocation example

```
lsnodecandidate
```

The resulting output:

```
id               panel_name UPS_serial_number UPS_unique_id    hardware serial_number product_mtm machine_signature
500507680C00003C KQ8FP4W                      500507680C00003C DH8      KQ8FP4W       9846-AC1    68CB-157E-45C4-02A1
```

lsnodedependentvdisks (Deprecated)

Attention: The **lsnodedependentvdisks** command is deprecated. Use the **lsdependentvdisks** command instead.

lsnodehw (SVC) / lsnodecanisterhw (Storwize family products)

Use the **lsnodehw** / **lsnodecanisterhw** command to display the configured and actual hardware configuration of nodes in the clustered system.

Syntax

```
►►— lsnodehw — | lsnodecanisterhw — ┬─────────┬┬────────────────────┬┬─ object_id ──┬──────────►◄
                                    └ -nohdr ┘└ -delim — delimiter ┘└─ object_name ┘
```

Parameters

-nohdr

(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*

(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim : on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

object_id | *object_name*

(Required) Specifies the object name or ID.

Description

Table 34 provides the possible values that are applicable to the attributes that are displayed as data in the output views.

Table 34. Attribute values for lsnodehw and lsnodecanisterhw

Attribute	Value
id	Indicates the node or node canister unique ID.
name	Indicates the node or node canister name.
status	Indicates the node or node canister status.
IO_group_id	Indicates the input/output (I/O) group ID.
IO_group_name	Indicates the I/O group name.
hardware	Indicates the hardware model, such as DH8.
actual_different	Indicates whether the node or node canisterhardware is different from the configured hardware.
actual_valid	Indicates whether the node or node canisterhardware is valid.
memory_configured	Indicates the configured amount of memory (in GB).
memory_actual	Indicates the currently installed amount of memory (in GB).
memory_valid	Indicates whether the actual memory is a valid configuration.
cpu_count	Indicates the maximum number of CPUs for the node.
cpu_socket	Indicates the ID of socket the CPU fields refer to.

Table 34. Attribute values for lsnodehw and lsnodecanisterhw (continued)

Attribute	Value
cpu_configured	Indicates the configured CPU for this socket.
cpu_actual	Indicates the currently installed CPU in this socket.
cpu_valid	Indicates whether the currently installed CPU is a valid configuration.
adapter_count	Indicates the maximum number of adapters for the node (differs by node type).
adapter_location	Indicates the location of this adapter.
adapter_configured	Indicates the configured adapter for this location.
adapter_actual	Indicates the currently installed adapter for this location.
adapter_valid	Indicates whether the adapter in this location is valid.
ports_different	Indicates whether the current hardware is able to provide more I/O ports? The values are yes and no.

An invocation example for SAN Volume Controller

```
lsnodehw -delim , 1
```

The resulting output:

```
id,1
name,hlcn114289
status,online
IO_group_id,0
IO_group_name,io_grp0
hardware,DH8
actual_different,yes
actual_valid,no
memory_configured,8
memory_actual,8
memory_valid,yes
cpu_count,2
cpu_socket,1
cpu_configured,4 core Intel(R) Xeon(R) CPU E3110 @ 3.0GHz
cpu_actual,4 core Intel(R) Xeon(R) CPU E3110 @ 3.0GHz
cpu_valid,yes
cpu_socket,2
cpu_configured,none
cpu_actual,none
cpu_valid,yes
adapter_count,4
adapter_location,0
adapter_configured,1Gb/s Ethernet adapter
adapter_actual,1Gb/s Ethernet adapter
adapter_valid,yes
adapter_location,0
adapter_configured,1Gb/s Ethernet adapter
adapter_actual,1Gb/s Ethernet adapter
adapter_valid,yes
adapter_location,1
adapter_configured,Four port 8Gb/s FC adapter card
adapter_actual,Four port 8Gb/s FC adapter card
adapter_valid,yes
adapter_location,2
adapter_configured,none
adapter_actual,Four port 8Gb/s FC adapter card
adapter_valid,no
ports_different yes
```

An invocation example for Storwize V7000

```
lsnodecanisterhw -delim , 1
```

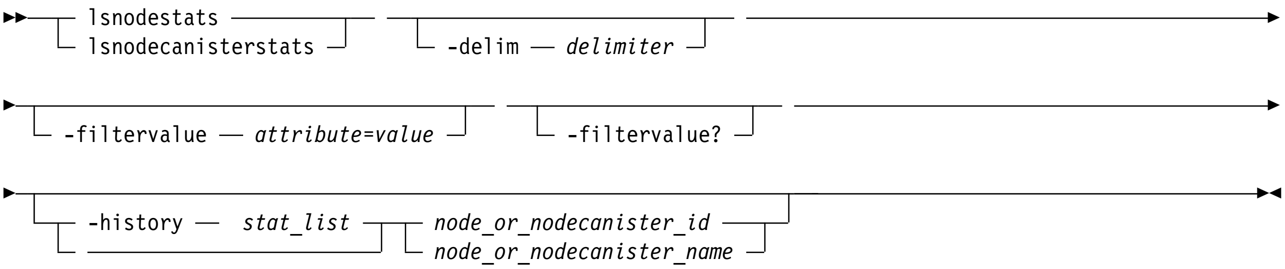
The resulting output

```
id,1
name,hlcn114289
status,online
IO_group_id,0
IO_group_name,io_grp0
hardware,112
...
```

lsnodestats (SVC) / lsnodecanisterstats (Storwize family products)

Use the **lsnodestats** / **lsnodecanisterstats** command to display the most recent values of statistics for all nodes or node canisters, and display all statistics for a particular node or node canister. Additionally, You can use this command to display a history of values for a given subset of available statistics.

Syntax



Parameters

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (**:**) separates all items of data in a concise view. (For example, the spacing of columns does not occur.) In a detailed view, the data is separated from its header by the specified delimiter.

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards:

- The wildcard character is the asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, enclose the filter entry within double quotation marks (""):

```
lsenclosurestats -filtervalue stat_name=temp_f
```

-filtervalue?
(Optional) Displays the valid filter attributes for the **-filtervalue** *attribute=value* parameter:

- node_id
- node_name

- stat_name

-history *stat_list*
(Optional) Provides a table of statistical values for the specified node. The *stat_list* is a colon-delimited list of one or more statistical values. A table is generated for each entry in the *stat_list*.

Remember: If **-history** is specified, a node ID or name must be specified as well.

node_or_nodecanister_id | *node_or_nodecanister_name*
(Optional) Identifies the node or node canister for which you want to request statistics.

Description

This command returns a concise list or a detailed view of nodes or node canisters that are part of the clustered system. Table 35 provides the possible values that are applicable to the attributes that are displayed as data in the output views.

Table 35. Attribute values for lsnodestats or lsnodecanister

Attribute	Value
node_id	The ID of the node or node canister.
node_name	The name of the node or node canister.
stat_current	The current value of the statistic field.
stat_list	The system history of the reported statistics. The list of statistics can contain multiple items that are separated by colons.
stat_name	The name of the statistic field. See Table 36 on page 236 for descriptions of available statistics.
stat_peak	The peak value of the statistic field in the last 5 minutes.
stat_peak_time	The time that the peak occurred.
sample_time	The time of the sample occurrence.
stat_value	The statistical value at the epoch interval.

Remember: Filtering is supported on the `stat_name` field by using the concise view.

An invocation example

```
lsnodestats
```

The resulting output:

```
node_id node_name stat_name      stat_current stat_peak stat_peak_time
1       node1     cpu_pc         5            9         111123105330
1       node1     fc_mb          218          238       111123105440
1       node1     fc_io          1122         1501      111123105435
1       node1     sas_mb         282          402       111123105335
1       node1     sas_io         3129         4427      111123105335
1       node1     iscsi_mb       0            0         111123105825
1       node1     iscsi_io       0            0         111123105825
1       node1     write_cache_pc 0            0         111123105825
1       node1     total_cache_pc 0            0         111123105825
1       node1     vdisk_mb       218          238       111123105440
1       node1     vdisk_io       1076         1452      111123105435
1       node1     vdisk_ms       52           60        111123105605
1       node1     mdisk_mb       218          238       111123105435
1       node1     mdisk_io       1874         2386      111123105435
1       node1     mdisk_ms       15           33        111123105605
1       node1     drive_mb       281          401       111123105335
1       node1     drive_io       3130         4060      111123105335
```

```
1       node1     drive_ms       13           27        111123105605
1       node1     vdisk_r_mb     134          157       111123105440
1       node1     vdisk_r_io     561          885       111123105430
1       node1     vdisk_r_ms     37           45        111123105605
1       node1     vdisk_w_mb     84           89        111123105700
1       node1     vdisk_w_io     515          587       111123105625
1       node1     vdisk_w_ms     67           84        111123105330
1       node1     mdisk_r_mb     133          155       111123105510
1       node1     mdisk_r_io     1337         1789      111123105435
1       node1     mdisk_r_ms     15           33        111123105605
1       node1     mdisk_w_mb     84           89        111123105700
1       node1     mdisk_w_io     536          611       111123105625
1       node1     mdisk_w_ms     17           32        111123105605
1       node1     drive_r_mb     151          295       111123105335
1       node1     drive_r_io     1700         2904      111123105335
1       node1     drive_r_ms     14           30        111123105605
1       node1     drive_w_mb     130          137       111123105700
1       node1     drive_w_io     1429         1586      111123105625
1       node1     drive_w_ms     12           22        111123105605
1       node1     iplink_mb      0            1         130523104536
1       node1     iplink_io      0            10        130523104536
2       node2     cpu_pc         6            7         111123105624
2       node2     fc_mb          132          145       111123105724
2       node2     fc_io          1519         1944      111123105739
2       node2     sas_mb         189          308       111123105619
2       node2     sas_io         2737         4099      111123105614
2       node2     iscsi_mb       0            0         111123105824
2       node2     iscsi_io       0            0         111123105824
2       node2     write_cache_pc 0            0         111123105824
2       node2     total_cache_pc 0            0         111123105824
2       node2     vdisk_mb       132          145       111123105724
2       node2     vdisk_io       1459         1892      111123105739
2       node2     vdisk_ms       47           81        111123105514
2       node2     mdisk_mb       132          145       111123105724
2       node2     mdisk_io       1635         2066      111123105739
2       node2     mdisk_ms       8            18        111123105619
2       node2     drive_mb       189          310       111123105619
2       node2     drive_io       2735         3750      111123105619
2       node2     drive_ms       9            20        111123105604
2       node2     vdisk_r_mb     20           21        111123105809
2       node2     vdisk_r_io     796          1180      111123105739
2       node2     vdisk_r_ms     2            8         111123105529
2       node2     vdisk_w_mb     112          134       111123105349
2       node2     vdisk_w_io     662          805       111123105504
2       node2     vdisk_w_ms     100          104       111123105624
2       node2     mdisk_r_mb     20           21        111123105809
2       node2     mdisk_r_io     951          1330      111123105739
2       node2     mdisk_r_ms     2            7         111123105529
2       node2     mdisk_w_mb     112          134       111123105349
2       node2     mdisk_w_io     684          834       111123105504
2       node2     mdisk_w_ms     16           36        111123105619
2       node2     drive_r_mb     17           132       111123105619
2       node2     drive_r_io     899          1920      111123105619
2       node2     drive_r_ms     6            12        111123105344
2       node2     drive_w_mb     171          206       111123105504
2       node2     drive_w_io     1837         2230      111123105504
2       node2     drive_w_ms     11           26        111123105619
1       node1     iplink_mb      0            1         130523104536
1       node1     iplink_io      0            10        130523104536
cloud_up_mb    0            0         161118051715
cloud_up_ms    0            0         161118051715
cloud_down_mb  0            0         161118051715
cloud_down_ms  0            0         161118051715
```

An invocation example of a node-based, filtered invocation

```
lsnodestats -filtervalue stat_name=sas_io:stat_name=sas_mb node1
```

The resulting output:

```
node_id node_name stat_name stat_current stat_peak stat_peak_time
1       node1     sas_mb    212          421       111123105840
1       node1     sas_io    2477         4184      111123105840
```

An invocation example of an historical view that can list multiple statistics and requires a node-based invocation

```
lsnodestats -history cpu_pc:fc_mb:sas_mb node1
```

The resulting output:

```
node_id node_name sample_time  stat_name stat_value
2       node2     111123105839 cpu_pc    6
2       node2     111123105844 cpu_pc    5
2       node2     111123105849 cpu_pc    5
2       node2     111123105854 cpu_pc    5
2       node2     111123105859 cpu_pc    6
2       node2     111123105904 cpu_pc    5
2       node2     111123105909 cpu_pc    5
2       node2     111123105914 cpu_pc    5
2       node2     111123105919 cpu_pc    5
2       node2     111123105924 cpu_pc    5
2       node2     111123105929 cpu_pc    5
2       node2     111123105934 cpu_pc    5
2       node2     111123105839 fc_mb     128
2       node2     111123105844 fc_mb     126
2       node2     111123105849 fc_mb     123
2       node2     111123105854 fc_mb     142
2       node2     111123105859 fc_mb     119
2       node2     111123105904 fc_mb     131
2       node2     111123105909 fc_mb     157
2       node2     111123105914 fc_mb     177
2       node2     111123105919 fc_mb     182
2       node2     111123105924 fc_mb     182
2       node2     111123105929 fc_mb     155
2       node2     111123105934 fc_mb     177
2       node2     111123105839 sas_mb    191
2       node2     111123105844 sas_mb    191
2       node2     111123105849 sas_mb    185
2       node2     111123105854 sas_mb    216
2       node2     111123105859 sas_mb    181
2       node2     111123105904 sas_mb    198
2       node2     111123105909 sas_mb    228
2       node2     111123105914 sas_mb    243
2       node2     111123105919 sas_mb    251
2       node2     111123105924 sas_mb    248
2       node2     111123105929 sas_mb    217
2       node2     111123105934 sas_mb    242
```

The following table provides the possible values that are applicable to the values that are displayed for the **stat_name** attribute.

Table 36. Stat_name field values

Value	Description
compression_cpu_pc	Displays the percentage of allocated CPU capacity that is utilized for compression.
cpu_pc	Displays the percentage of allocated CPU capacity that is utilized for the system.

Table 36. Stat_name field values (continued)

Value	Description
fc_mb	Displays the total number of megabytes transferred per second for Fibre Channel traffic on the system. This value includes host I/O and any bandwidth that is used for communication within the system.
fc_io	Displays the total input/output (I/O) operations that are transferred per seconds for Fibre Channel traffic on the system. This value includes host I/O and any bandwidth that is used for communication within the system.
sas_mb	Displays the total number of megabytes transferred per second for serial-attached SCSI (SAS) traffic on the system. This value includes host I/O and bandwidth that is used for background RAID activity.
sas_io	Displays the total I/O operations that are transferred per second for SAS traffic on the system. This value includes host I/O and bandwidth that is used for background RAID activity.
iscsi_mb	Displays the total number of megabytes transferred per second for iSCSI traffic on the system.
iscsi_io	Displays the total I/O operations that are transferred per second for iSCSI traffic on the system.
write_cache_pc	Displays the percentage of the write cache usage for the node.
total_cache_pc	Displays the total percentage for both the write and read cache usage for the node.
vdisk_mb	Displays the average number of megabytes transferred per second for read and write operations to volumes during the sample period.
vdisk_io	Displays the average number of I/O operations that are transferred per second for read and write operations to volumes during the sample period.
vdisk_ms	Displays the average amount of time in milliseconds that the system takes to respond to read and write requests to volumes over the sample period.
mdisk_mb	Displays the average number of megabytes transferred per second for read and write operations to MDisks during the sample period.
mdisk_io	Displays the average number of I/O operations that are transferred per second for read and write operations to MDisks during the sample period.
mdisk_ms	Displays the average amount of time in milliseconds that the system takes to respond to read and write requests to MDisks over the sample period.
drive_mb	Displays the average number of megabytes transferred per second for read and write operations to drives during the sample period
drive_io	Displays the average number of I/O operations that are transferred per second for read and write operations to drives during the sample period.
drive_ms	Displays the average amount of time in milliseconds that the system takes to respond to read and write requests to drives over the sample period.
vdisk_w_mb	Displays the average number of megabytes transferred per second for read and write operations to volumes during the sample period.
vdisk_w_io	Displays the average number of I/O operations that are transferred per second for write operations to volumes during the sample period.
vdisk_w_ms	Displays the average amount of time in milliseconds that the system takes to respond to write requests to volumes over the sample period.
mdisk_w_mb	Displays the average number of megabytes transferred per second for write operations to MDisks during the sample period.
mdisk_w_io	Displays the average number of I/O operations that are transferred per second for write operations to MDisks during the sample period.
mdisk_w_ms	Displays the average amount of time in milliseconds that the system takes to respond to write requests to MDisks over the sample period.

Table 36. Stat_name field values (continued)

Value	Description
drive_w_mb	Displays the average number of megabytes transferred per second for write operations to drives during the sample period
drive_w_io	Displays the average number of I/O operations that are transferred per second for write operations to drives during the sample period.
drive_w_ms	Displays the average amount of time in milliseconds that the system takes to respond write requests to drives over the sample period.
vdisk_r_mb	Displays the average number of megabytes transferred per second for read operations to volumes during the sample period.
vdisk_r_io	Displays the average number of I/O operations that are transferred per second for read operations to volumes during the sample period.
vdisk_r_ms	Displays the average amount of time in milliseconds that the system takes to respond to read requests to volumes over the sample period.
mdisk_r_mb	Displays the average number of megabytes transferred per second for read operations to MDisks during the sample period.
mdisk_r_io	Displays the average number of I/O operations that are transferred per second for read operations to MDisks during the sample period.
mdisk_r_ms	Displays the average amount of time in milliseconds that the system takes to respond to read requests to MDisks over the sample period.
drive_r_mb	Displays the average number of megabytes transferred per second for read operations to drives during the sample period
drive_r_io	Displays the average number of I/O operations that are transferred per second for read operations to drives during the sample period.
drive_r_ms	Displays the average amount of time in milliseconds that the system takes to respond to read requests to drives over the sample period.
iplink_mb	The total number of megabytes transferred per second for Internet Protocol (IP) replication traffic on the system. This value does not include iSCSI host input/output (I/O) operations.
iplink_comp_mb	Displays the average number of compressed megabytes transferred per second (MBps) over the IP replication link during the sample period.
iplink_io	The total input/output (I/O) operations that are transferred per second for IP partnership traffic on the system. This value does not include Internet Small Computer System Interface (iSCSI) host I/O operations.
cloud_up_mb	Displays the average number of megabits transferred per second (Mbps) for upload operations to a cloud account during the sample period.
cloud_up_ms	Displays the average amount of time (in milliseconds) it takes for the system to respond to upload requests to a cloud account during the sample period.
cloud_down_mb	Displays the average number of Mbps for download operations to a cloud account during the sample period.
cloud_down_ms	Displays the average amount of time (in milliseconds) it takes for the system to respond to download requests to a cloud account during the sample period.

An invocation example

```
lsnodecanisterstats
```

The resulting output:

```
node_id node_name stat_name      stat_current stat_peak stat_peak_time
1       node1     cpu_pc         5            9         111123105330
1       node1     fc_mb          218          238       111123105440
```

```
1       node1     fc_io          1122         1501      111123105435
1       node1     sas_mb         282          402       111123105335
1       node1     sas_io         3129         4427      111123105335
1       node1     iscsi_mb       0            0         111123105825
1       node1     iscsi_io       0            0         111123105825
1       node1     write_cache_pc 0            0         111123105825
1       node1     total_cache_pc 0            0         111123105825
1       node1     vdisk_mb       218          238       111123105440
1       node1     vdisk_io       1076         1452      111123105435
1       node1     vdisk_ms       52           60        111123105605
1       node1     mdisk_mb       218          238       111123105435
1       node1     mdisk_io       1874         2386      111123105435
1       node1     mdisk_ms       15           33        111123105605
1       node1     drive_mb       281          401       111123105335
1       node1     drive_io       3130         4060      111123105335
1       node1     drive_ms       13           27        111123105605
1       node1     vdisk_r_mb     134          157       111123105440
1       node1     vdisk_r_io     561          885       111123105430
1       node1     vdisk_r_ms     37           45        111123105605
1       node1     vdisk_w_mb     84           89        111123105700
1       node1     vdisk_w_io     515          587       111123105625
1       node1     vdisk_w_ms     67           84        111123105330
1       node1     mdisk_r_mb     133          155       111123105510
1       node1     mdisk_r_io     1337         1789      111123105435
1       node1     mdisk_r_ms     15           33        111123105605
1       node1     mdisk_w_mb     84           89        111123105700
1       node1     mdisk_w_io     536          611       111123105625
1       node1     mdisk_w_ms     17           32        111123105605
1       node1     drive_r_mb     151          295       111123105335
1       node1     drive_r_io     1700         2904      111123105335
1       node1     drive_r_ms     14           30        111123105605
1       node1     drive_w_mb     130          137       111123105700
1       node1     drive_w_io     1429         1586      111123105625
1       node1     drive_w_ms     12           22        111123105605
1       node1     iplink_mb      0            1         130523104536
1       node1     iplink_io      0            10        130523104536
2       node2     cpu_pc         6            7         111123105624
2       node2     fc_mb          132          145       111123105724
2       node2     fc_io          1519         1944      111123105739
2       node2     sas_mb         189          308       111123105619
2       node2     sas_io         2737         4099      111123105614
2       node2     iscsi_mb       0            0         111123105824
2       node2     iscsi_io       0            0         111123105824
2       node2     write_cache_pc 0            0         111123105824
2       node2     total_cache_pc 0            0         111123105824
2       node2     vdisk_mb       132          145       111123105724
2       node2     vdisk_io       1459         1892      111123105739
2       node2     vdisk_ms       47           81        111123105514
2       node2     mdisk_mb       132          145       111123105724
2       node2     mdisk_io       1635         2066      111123105739
2       node2     mdisk_ms       8            18        111123105619
2       node2     drive_mb       189          310       111123105619
2       node2     drive_io       2735         3750      111123105619
2       node2     drive_ms       9            20        111123105604
2       node2     vdisk_r_mb     20           21        111123105809
2       node2     vdisk_r_io     796          1180      111123105739
2       node2     vdisk_r_ms     2            8         111123105529
2       node2     vdisk_w_mb     112          134       111123105349
2       node2     vdisk_w_io     662          805       111123105504
2       node2     vdisk_w_ms     100          104       111123105624
2       node2     mdisk_r_mb     20           21        111123105809
2       node2     mdisk_r_io     951          1330      111123105739
2       node2     mdisk_r_ms     2            7         111123105529
2       node2     mdisk_w_mb     112          134       111123105349
2       node2     mdisk_w_io     684          834       111123105504
2       node2     mdisk_w_ms     16           36        111123105619
2       node2     drive_r_mb     17           132       111123105619
```

```
2       node2     drive_r_io     899         1920      111123105619
2       node2     drive_r_ms     6           12        111123105344
2       node2     drive_w_mb     171         206       111123105504
2       node2     drive_w_io     1837        2230      111123105504
2       node2     drive_w_ms     11          26        111123105619
1       node1     iplink_mb      0           1         130523104536
1       node1     iplink_io      0           10        130523104536
iplink_comp_mb      0           250        151014133723
```

An invocation example of a node-based, filtered invocation example for lsnodecanisterstats

```
lsnodecanisterstats -filtervalue stat_name=sas_io:stat_name=sas_mb node1
```

The resulting output:

```
node_id node_name stat_name stat_current stat_peak stat_peak_time
1       node1     sas_mb    212          421       111123105840
1       node1     sas_io    2477         4184      111123105840
```

An invocation example of a historical view that can list multiple statistics and requires a node-based invocation

```
lsnodecanisterstats -history cpu_pc:fc_mb:sas_mb node1
```

The resulting output:

```
node_id node_name sample_time  stat_name stat_value
2       node2     111123105839 cpu_pc    6
2       node2     111123105844 cpu_pc    5
2       node2     111123105849 cpu_pc    5
2       node2     111123105854 cpu_pc    5
2       node2     111123105859 cpu_pc    6
2       node2     111123105904 cpu_pc    5
2       node2     111123105909 cpu_pc    5
2       node2     111123105914 cpu_pc    5
2       node2     111123105919 cpu_pc    5
2       node2     111123105924 cpu_pc    5
2       node2     111123105929 cpu_pc    5
2       node2     111123105934 cpu_pc    5
2       node2     111123105839 fc_mb     128
2       node2     111123105844 fc_mb     126
2       node2     111123105849 fc_mb     123
2       node2     111123105854 fc_mb     142
2       node2     111123105859 fc_mb     119
2       node2     111123105904 fc_mb     131
2       node2     111123105909 fc_mb     157
2       node2     111123105914 fc_mb     177
2       node2     111123105919 fc_mb     182
2       node2     111123105924 fc_mb     182
2       node2     111123105929 fc_mb     155
2       node2     111123105934 fc_mb     177
2       node2     111123105839 sas_mb    191
2       node2     111123105844 sas_mb    191
2       node2     111123105849 sas_mb    185
2       node2     111123105854 sas_mb    216
2       node2     111123105859 sas_mb    181
2       node2     111123105904 sas_mb    198
2       node2     111123105909 sas_mb    228
2       node2     111123105914 sas_mb    243
2       node2     111123105919 sas_mb    251
2       node2     111123105924 sas_mb    248
2       node2     111123105929 sas_mb    217
2       node2     111123105934 sas_mb    242
```

lsnodevpd (SVC) / lsnodecanistervpd (Storwize family products)

Use the **lsnodevpd** / **lsnodecanistervpd** command to display the vital product data (VPD) for each node.

Syntax

```
►►— lsnodevpd — | lsnodecanistervpd — ┬───────────┬─ ┬───────────────────────┬─ ──────────►
                                      └─ -nohdr ──┘   └─ -delim — delimiter ──┘

►─┬─ object_id ───┬──────────────────────────────────────────────────────────────────►◄
  └─ object_name ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. Using the **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

object_id | *object_name*
(Required) Specifies the object name or ID.

Description

This command displays the VPD for the specified node or node canister. Each field is reported on a new line. All fields are strings. The VPD is split into sections. Each section has a section heading. The number of fields in that section follows each heading. Each section is separated by an empty line.

For example:

```
section name:3 fields
field1:value
field2:value
field3:value

new section:x fields
...
```

Some sections contain information about multiple objects of that type. Each object within the section is separated by an empty line.

For example:

```
section name:4 fields
object1 field1:value
object1 field2:value
```

```
object2 field1:value
object2 field2:value

new section: x fields
...
```

Note: For SAN Volume Controller 2145-8G4 nodes, the VPD displays the device serial number of the Fibre Channel adapter as N/A.

Table 37. Attribute values for lsnodevpd and lsnodecanistervpd

Value	Description
system board	Displays the system board information.
part_number	Displays the total number of system part numbers.
system_serial_number	Displays the total number of system serial numbers.
number_of_processors	Displays the total number of system processors.
number_of_memory_modules	Displays the total number of memory modules.
number_of_fans	Displays the total number of system fans.
number_of_FC_cards	Displays the total number of Fibre Channel (FC) cards.
number_of_Ethernet_cards	Displays the total number of Ethernet cards.
iscsi_initiator_name	Displays the iSCSI IQN that is stored in node vital product data (VPD).

An invocation example for SAN Volume Controller

```
lsnodevpd 1
```

The resulting output:

```
id 1
system board: 21 fields
part_number 43V7072
system_serial_number KD1438A
number_of_processors 4
number_of_memory_modules 6
number_of_fans 6
number_of_generic_devices 3
number_of_FC_adapters 1
number_of_Ethernet adapters 3
number_of_SAS_adapters 0
number_of_Bus_adapters 0
number_of_power_supplies 2
number_of_local_managed_disks 0
BIOS_manufacturer IBM Corp.
BIOS_version -[D6E124AUS-1.01]-
BIOS_release_date 04/30/2009
system_manufacturer IBM
system_product System x3650 M4 -[2145DH8]-
version 00
system_product IBM System x -[2145DH8]-
planar_manufacturer IBM
CMOS_battery_part_number 33F8354
frame_assembly_part_number
power_cable_assembly_part_number 31P1294
service_processor_firmware 1.01
disk_controller 44E8690

processor: 6 fields
part_number 46D1266
processor_location Processor 1
```

```
manufacturer Intel(R) Corporation
version Intel(R) Xeon(R) CPU           E5530  @ 2.40GHz
speed 2400
status Enabled
memory module: 96 fields
part_number 44T1493
device_location DIMM01
bank_location BANK01
size (MB) No Module Installed
manufacturer Not Specified
serial_number Not Specified

part_number 44T1493
device_location DIMM02
bank_location BANK02
size (MB) 4096
manufacturer Samsung
serial_number 99062848

part_number 44T1493
device_location DIMM03
bank_location BANK03
size (MB) 4096
manufacturer Samsung
serial_number C7062848

part_number 44T1493
device_location DIMM04
bank_location BANK04
size (MB) No Module Installed
manufacturer Not Specified
serial_number Not Specified

part_number 44T1493
device_location DIMM05
bank_location BANK05
size (MB) 4096
manufacturer Hynix
serial_number 12F41112

part_number 44T1493
device_location DIMM06
bank_location BANK06
size (MB) 4096
manufacturer Hynix
serial_number 2AF41112

part_number 44T1493
device_location DIMM07
bank_location BANK07
size (MB) 4096
manufacturer Hynix
serial_number D128312E

part_number 44T1493
device_location DIMM08
bank_location BANK08
size (MB) 4096
manufacturer Hynix
serial_number D028C12E

part_number 44T1493
device_location DIMM09
bank_location BANK09
size (MB) No Module Installed
manufacturer Not Specified
serial_number Not Specified
```

```
part_number 44T1493
device_location DIMM10
bank_location BANK10
size (MB) No Module Installed
manufacturer Not Specified
serial_number Not Specified

part_number 44T1493
device_location DIMM11
bank_location BANK11
size (MB) No Module Installed
manufacturer Not Specified
serial_number Not Specified

part_number 44T1493
device_location DIMM12
bank_location BANK12
size (MB) No Module Installed
manufacturer Not Specified
serial_number Not Specified

part_number 44T1493
device_location DIMM13
bank_location BANK13
size (MB) No Module Installed
manufacturer Not Specified
serial_number Not Specified

part_number 44T1493
device_location DIMM14
bank_location BANK14
size (MB) No Module Installed
manufacturer Not Specified
serial_number Not Specified

part_number 44T1493
device_location DIMM15
bank_location BANK15
size (MB) No Module Installed
manufacturer Not Specified
serial_number Not Specified

part_number 44T1493
device_location DIMM16
bank_location BANK16
size (MB) No Module Installed
manufacturer Not Specified
serial_number Not Specified

fan: 12 fields
part_number 43V6929
location location1

part_number 43V6929
location location2

part_number 43V6929
location location3

part_number 43V6929
location location4

part_number 43V6929
location location5

part_number 43V6929
```

```
location location6

Adapter card: 18 fields
card_type FC card
part_number 31P1337
port_numbers 1 2 3 4
location 0
device_serial_number 11S31P1333YM10MY96A206
manufacturer IBM
device QE8
card_revision 2
chip_revision 2.0

Fibre channel port: 44 fields
part_number 31P1338
manufacturer JDSU
device PLRXPLVCSH423N
serial_number C945VK0RB
supported_speeds 2,4,8 Gbps
connector_type LC
transmitter_type SN
wavelength 850
max_distance_by_cable_type OM1:20,OM2:50,OM3:150
hw_revision 2
port_number 1

part_number 31P1338
manufacturer JDSU
device PLRXPLVCSH423N
serial_number C945VK0KU
supported_speeds 2,4,8 Gbps
connector_type LC
transmitter_type SN
wavelength 850
max_distance_by_cable_type OM1:20,OM2:50,OM3:150
hw_revision 2
port_number 2

part_number 31P1338
manufacturer JDSU
device PLRXPLVCSH423N
serial_number C945VK0KT
supported_speeds 2,4,8 Gbps
connector_type LC
transmitter_type SN
wavelength 850
max_distance_by_cable_type OM1:20,OM2:50,OM3:150
hw_revision 2
port_number 3

part_number 31P1338
manufacturer JDSU
device PLRXPLVCSH423N
serial_number C945VK0RA
supported_speeds 2,4,8 Gbps
connector_type LC
transmitter_type SN
wavelength 850
max_distance_by_cable_type OM1:20,OM2:50,OM3:150
hw_revision 2
port_number 4

Adapter card: 9 fields
card_type Ethernet
part_number 43V7072
port_numbers 1 2
location 0
```

```
device_serial_number 0123456789
manufacturer Unknown
device NetXtreme II Gigabit Ethernet
card_revision Unknown
chip_revision 8.0

Ethernet port: 22 fields
part_number Unknown
manufacturer N/A
device N/A
serial_number N/A
supported_speeds 10,100 Mbps,1 Gbps
connector_type N/A
transmitter_type N/A
wavelength N/A
max_distance_by_cable_type N/A
hw_revision N/A
port_number 1

part_number Unknown
manufacturer N/A
device N/A
serial_number N/A
supported_speeds 10,100 Mbps,1 Gbps
connector_type N/A
transmitter_type N/A
wavelength N/A
max_distance_by_cable_type N/A
hw_revision N/A
port_number 2

Adapter card: 9 fields
card_type Ethernet
part_number 31P1559
port_numbers 3 4
location 2
device_serial_number BT05149496
manufacturer Emulex Corp
device Emulex/OneConnect 10Gb NIC (be3)
card_revision 1.0
chip_revision 0.2

Ethernet port: 22 fields
part_number 31P1549
manufacturer FINISAR CORP.
device FTLX8571D3BCL
serial_number AHE05K7
supported_speeds 10 Gbps
connector_type LC
transmitter_type 10G Base-SR
wavelength 850
max_distance_by_cable_type OM1:30,OM2:80,OM3:300
hw_revision A
port_number 3

part_number 31P1549
manufacturer JDSU
device PLRXPLSCS4321N
serial_number C825UB0D2
supported_speeds 10 Gbps
connector_type LC
transmitter_type 10G Base-SR
wavelength 850
max_distance_by_cable_type OM1:30,OM2:80,OM3:300
hw_revision 1
port_number 4
```

```
device: 24 fields
part_number 31P1339
bus USB
device 0
model IBM USB Endeavour
revision 1.1
serial_number NA
approx_capacity 0
hw_revision 0

part_number 42D0673
bus scsi
device 0
model MBE2073RC
revision SC13
serial_number D3A01C0HSC13SC13SC1
approx_capacity 68
hw_revision

part_number N/A
bus scsi
device 0
model STEC USB 2.0
revision 1113
serial_number NA
approx_capacity 1
hw_revision

system code level: 4 fields
id 58
node_name dvt151769
WWNN 0x500507680100b7d2
code_level 6.4.1.3 (build 75.0.1212193000)
object_name_model

front panel assembly: 3 fields
front_panel_id 151769

part_number N/A

battery_midplane_FRU_part 12Z9880
battery_midplane_part_identity 11S98Z1230YM11RM234567
battery_midplane_FW_version 1.6
battery_power_cable_FRU_part 12Z9881
battery_power_sense_cable_FRU_part 12Z9882
battery_comms_cable_FRU_part 12Z9883
battery_EPOW_cable_FRU_part 12Z9884

iscsi_initiator_name iqn.2009-05.cloud.com:test.node1
```

An invocation example for Storwize V7000

```
lsnodecanistervpd 1
```

The resulting output:

```
id 1

system board: 21 fields
part_number 43V7072
system_serial_number KD1438A
number_of_processors 4
number_of_memory_modules 6
number_of_fans 6
number_of_FC_cards 1
```

```
number_of_Ethernet cards 3
number_of_scsi/ide_devices 2
BIOS_manufacturer IBM Corp.
BIOS_version -[D6E124AUS-1.01]-
BIOS_release_date 04/30/2009
system_manufacturer IBM
system_product System x3650 M4 -[2145DH8]-
version 00planar_manufacturer IBM
planar_product 49Y6498
planar_version (none)
power_supply_part_number 39Y7201
CMOS_battery_part_number 33F8354
frame_assembly_part_number
ethernet_cable_part_number
service_processor_firmware 1.01

processor: 6 fields
processor_location Processor 1
manufacturer Intel(R) Corporation
version Intel(R) Xeon(R) CPU           E5530  @ 2.40GHz
speed 2400
status Enabled
CPU_part_number 46D1266

memory module: 96 fields
part_number 44T1493
device_location DIMM01
bank_location BANK01
size (MB) No Module Installed
manufacturer Not Specified
serial_number Not Specified

part_number 44T1493
device_location DIMM02
bank_location BANK02
size (MB) 4096
manufacturer Samsung
serial_number 99062848

part_number 44T1493
device_location DIMM03
bank_location BANK03
size (MB) 4096
manufacturer Samsung
serial_number C7062848
...

fan: 12 fields
part_number 43V6929
location location1

part_number 43V6929
location location2

part_number 43V6929
location location3
...

Adapter card: 18 fields
card_type FC card
part_number 31P1337
port_numbers 1 2 3 4
location 0
device_serial_number 11S31P1333YM10MY96A206
manufacturer IBM
device QE8
card_revision 2
```

```
chip_revision 2.0

card_type SAS card
part_number 44E8690
port_numbers 1 2 3 4
location 0
device_serial_number 11S31P1299YM10MY948004
manufacturer IBMHUR
device Capri-PMC8001
card_revision Y
chip_revision 1.1

Fibre Channel SFP: 48 fields
part_number 17P9211
manufacturer JDSU
device PLRXPLVCSH4921
serial_number C915EB06V
supported_speeds 2,4,8
connector_type LC
transmitter_type SN
wavelength 850
max_distance_by_cable_type OM1:20,OM2:50,OM3:150
hw_revision 1
port_number 1
WWPN 500507680140350d
...

device: 15 fields
part_number 31P1339
bus USB
device 0
model IBM USB Endeavour
revision 1.0
serial_number NA
approx_capacity 0
hw_revision 0

part_number 42D0673
bus scsi
device 0
model ST973452SS
revision B623
serial_number 3TA00BZ20109B623
approx_capacity 68

software: 8 fields
code_level 5.1.0.0 (build 16.1.0906240000)
nodecanister_name nodecanister1
ethernet_status 1

ethernet_status 0
WWNN 0x500507680100350d
id 1
MAC_address 00 21 5e 09 09 08
MAC_address 00 21 5e 09 09 0a

front panel assembly: 3 fields
front_panel_id 161040
front_panel_locale en_US

part_number N/A

UPS: 10 fields
electronics_assembly_part_number 64P8326
battery_part_number 31P0710
battery: 7 fields
battery_midplane_FRU_part 12Z9880
```

```
battery_midplane_part_identity 11S98Z1230YM11RM234567
battery_midplane_FW_version 1.6
battery_power_cable_FRU_part 12Z9881
battery_power_sense_cable_FRU_part 12Z9882
battery_comms_cable_FRU_part 12Z9883
battery_EPOW_cable_FRU_part 12Z9884
UPS_assembly_part_number 64P8326
input_power_cable_part_number CountryDependent
UPS_serial_number 1000840050
UPS_type 2145UPS 1U
UPS_internal_part_number P31P0875
UPS_unique_id 0x20400002047c0140
UPS_main_firmware 1.02
UPS_comms_firmware 1.20
iscsi_initiator_name iqn.2009-05.cloud.com:test.node1
```

lsportusb

Use the **lsportusb** command to display information about Universal Serial Bus (USB) ports.

Syntax

```
►►── lsportusb ──┬──────────┬── ┬──────────────────────┬─┬───────────────┬──────────────►◄
                 └─ -nohdr ─┘   └─ -delim ── delimiter ─┘ └─ usb_port_id ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim **:** on the command line, the colon character (**:**) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

usb_port_id
(Optional) Specifies the USB port ID. Used when detailed information about a USB port is required.

Description

This command displays information about Universal Serial Bus (USB) ports.

Table 38 provides the attribute values that can be displayed as output view data.

Table 38. lsportusb output

Attribute	**Possible Values**
id	Indicates the unique ID of the USB port in the system. This ID is the usb_port_id. The value is a numeric 0 or greater.

Table 38. lsportusb output (continued)

Attribute	**Possible Values**
node_id	Indicates the ID of the node where the USB port is. The value is a numeric string.
node_name	Indicates the name of the node where the USB port is. The value is an alphanumeric string.
node_side	Indicates the side of the node where the USB port is. The values are `front` and `rear`.
port_id	Indicates the ID of the USB port on the node side. The value is a numeric 1 or greater.
status	Indicates the status of the USB port. The values are: • `active`, which indicates that a USB flash drive is plugged in and can be used by the system. • `inactive`, which indicates that no USB flash drive is detected. • `unsupported`, which indicates that a USB device is plugged in but cannot be used.
encryption_state	Indicates the encryption status of the USB device that is attached to the port. The values are: • Blank indicates that it is not in use for encryption • `validated` indicates that encryption keys are present and validated • `missing` indicates that encryption keys were validated and were then removed, and the DMP must run to confirm the absence. • `prepared` indicates that encryption keys are prepared as part of a rekey operation. • `validated_prepared` indicates that encryption keys are validated and prepared as part of a rekey operation. • `wrong_system` indicates that encryption keys are detected on the USB device but none valid for the system. • `old` indicates that the USB device contains encryption keys that were generated for this system - but they are not the current keys. • `error` indicates that an encryption key is detected and something might be wrong with it.
encryption_filename	Indicates the name of the file in the rot directory of the USB device to which the encryption state relates. The file name can contain up to 110 characters.
service_state	Indicates the USB command status. The values are: • Blank indicates that no command is active. • `running` indicates that `satask.txt` is processing, and default USB processing is ongoing. • `complete` indicates that `satask.txt` is processing and default USB processing is complete. • `install_image` indicates that `satask.txt` processing cannot start because there is an installation image on the USB flash drive.

A concise invocation example

```
lsportusb
```

The resulting output:

```
id:node_id:node_name:node_side:port_id:status:service_state
0:1:node1:rear:1:inactive
1:1:node1:rear:2:active:validated:complete
2:2:node2:rear:1:active::complete
3:2:node2:rear:2:active:wrong_system:complete
```

A detailed invocation example

```
lsportusb 3
```

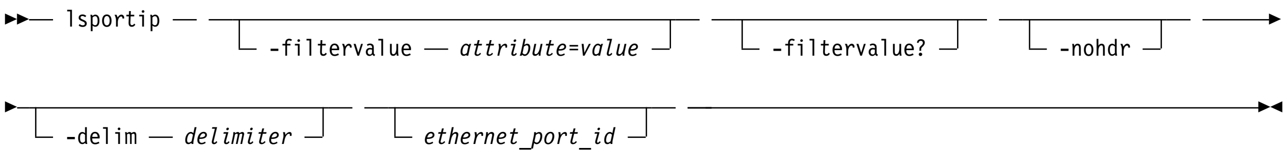
The resulting output:

```
id 3
node_id 2
node_name node2
node_side rear
port_id 2
status active

encryption_state wrong_system
encryption_filename encryption_key_filename_BadSystem
service_state complete
```

lsportip

Use the **lsportip** command to list the configuration for each Ethernet port on each node in the system. This command shows the Internet Protocol (IP) address and whether the port is configured as an Internet Small Computer Systems Interface (iSCSI) port.

Syntax



Parameters

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are returned. If a capacity is specified, the units must also be included.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards with the SAN Volume Controller CLI:

- The wildcard character is an asterisk (*).
- The command can contain a maximum of one wildcard, which must be the first or last character in the string.
- When you use a wildcard, enclose the filter entry within double quotation marks (""), as follows:

```
lsportip -filtervalue "node_name=md*"
```

-filtervalue?
(Optional) Displays the valid filter attributes. The following filter attributes for the **lsportip** command are valid:

- id
- node_id
- node_name
- state
- failover

-nohdr

(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*

(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

ethernet_port_id

(Optional) Specifies the ID of an Ethernet port (`1`, `2`, `3`, or `4`). If omitted, a concise view is displayed for all ports. When you use this parameter, the detailed view of the specified port is returned and any value that is specified by the **-filtervalue** parameter is ignored. If you do not use the *ethernet_port_id* parameter, the concise view displays all ports that match the filtering requirements that are specified by the **-filtervalue** parameter.

Description

This command lists the configuration of the Ethernet ports for each node in the IBM Spectrum Virtualize system.

Use the **lsportip** command with the optional **ethernet_port_id** parameter to display a detailed view of the specified port.

Output rows for a port show the MAC address of that port if it can be determined. If the node and the Ethernet link are online, the rows also show the speed and duplex state of the link. The duplex field can have values of `Half` or `Full`, or it is blank if the node is offline.

The fourth row for each port shows any IP address that is configured for that port and are not failed over to a different node. The failover field on this row is set to `no`. The second row for each port shows any iSCSI addresses that are configured for the partner node, or for the local node with failover, and that are active on the port. The failover field on this row is set to `yes`.

The state field is set to `unconfigured` if there are no iSCSI addresses that are configured on the port. The state field is set to `offline` if there are configured addresses but the link is down, and `online` if the link is up. Any offline rows indicate a potential problem.

This command displays information about system port status.

Table 39 on page 254 provides the attribute values that can be displayed as output view data.

Table 39. lsportip output

Attribute	Description
id	Indicates the ID of the Ethernet port.
node_id	Indicates the ID of the node that contains the port.
node_name	Indicates the name of the node that contains the port.
IP address	Indicates the IPv4 address (and is blank if there is none).
mask	Indicates the IPv4 subnet mask (and is blank if there is none).
gateway	Indicates the IPv4 gateway (and is blank if there is none).
IP_address_6	Indicates the IPv6 address (and is blank if there is none).
prefix_6	Indicates the IPv6 prefix (and is blank if there is none).
gateway_6	Indicates the IPv6 gateway address (and is blank if there is none).
MAC	Indicates the current MAC address (blank if unknown).
duplex	Indicates the current duplex state of the port (blank if unknown).
state	Indicates the state of iSCSI addresses. The values can be: • `unconfigured`: There is no iSCSI address (or hardware might not exist). • `configured`: The iSCSI address is configured. • `management_only`: It is not configurable for I/O operations.
link_state	Indicates the link state of Ethernet port. The values are `active` and `inactive`.
host	Displays the IPv4 address that is used for host attach.
remote_copy	Displays the IPv4 remote copy port group ID. Remote copy includes Metro Mirror, Global Mirror, and HyperSwap.
host_6	Displays the IPv6 address that is used for host attach.
remote_copy_6	Displays the IPv6 remote copy port group ID. Remote copy includes Metro Mirror, Global Mirror, and HyperSwap.
remote_copy_status	Displays the IPv4 remote copy status. Remote copy includes Metro Mirror, Global Mirror, and HyperSwap.
remote_copy_status_6	Displays the IPv6 remote copy status. Remote copy includes Metro Mirror, Global Mirror, and HyperSwap.
vlan	Displays the virtual local area network (VLAN) ID associated with the IPv4 address on this port (a numeric character in the range 1 - 4094).
vlan_6	Displays the VLAN ID associated with the IPv6 address on this port (a numeric character from 1- 4094).
adapter_location	Displays the location of the adaptor that contains the Ethernet port (any number in the range 0 - 8). Where 1 - 8 is the PCIe expansion slot number and 0 means that the adaptor is part of the system board or not in a PCIe expansion slot.
adapter_port_id	Displays the location of the Ethernet port that is in the adapter (any number in the range 1 - 4).

Table 39. lsportip output (continued)

Attribute	Description
dcbx_state	Displays the DCBx state of the port. A value of: • `unsupported` indicates that the port does not accept Priority Flow Control (PFC) configuration from the switch port, even if the switch is DCBx-capable. All ports that are 1 Gbps have this value. • `enabled` indicates that the connected switch port is enabled for DCBx and the port state is online. • `disabled` indicates that the connected switch port turned off for DCBx or the port state is offline. On Ethernet ports that are 10 Gb/s, DCBx is automatically enabled if the connected switch port has it enabled. **Remember:** When this field is disabled or unsupported, all fields other than `lossless_iscsi` and `lossless_iscsi6` are blank.
iscsi_priority_tag	Displays the numeric priority tag value for the Internet Small Computer System Interface (iSCSI) protocol that is assigned on the connected switch port. This priority value must be a number from `0` to `7` or blank.
fcoe_priority_tag	Displays the numeric priority tag value for the Fiber Channel over Ethernet (FCoE) protocol that is assigned on the connected switch port. This value must be a number from `0` to `7` or blank.
pfc_enabled_tags	Displays a list of priority tags for which PFC is enabled on the connected switch port. If you want to use the lossless iSCSI or FCoE function, PFC must be enabled for the corresponding tags on the switch. Once enabled on the switch, the tags are displayed in this field. This value is either blank or is a colon-separated list of numbers from `0` to `7`. **Important:** If no priority tags are defined on the switch, this field is blank. If priority tags are defined on the switch but PFC is not enabled for those priority tags, this field is blank.
priority_group_0	Displays the set of priority tags that are within the priority group zero. This value is either blank or is a colon-separated list of numbers from `0` to `7`. This field is part of the Enhanced Transmission Selection (ETS) settings.
priority_group_1	Displays the set of priority tags that are within the priority group one. This value is either blank or is a colon-separated list of numbers from `0` to `7`. This field is part of the ETS settings.
priority_group_2	Displays the set of priority tags that are within the priority group two. This value is either blank or is a colon-separated list of numbers from `0` to `7`. This field is part of the ETS settings.
priority_group_3	Displays the set of priority tags that are within the priority group three. This value is either blank or is a colon-separated list of numbers from `0` to `7`. This field is part of the ETS settings.
priority_group_4	Displays the set of priority tags that are within the priority group four. This value is either blank or is a colon-separated list of numbers from `0` to `7`. This field is part of the ETS settings.
priority_group_5	Displays the set of priority tags that are within the priority group five. This value is either blank or is a colon-separated list of numbers from `0` to `7`. This field is part of the ETS settings.
priority_group_6	Displays the set of priority tags that are within the priority group six. This value is either blank or is a colon-separated list of numbers from `0` to `7`. This field is part of the ETS settings.
priority_group_7	Displays the set of priority tags that are within the priority group seven. This value is either blank or is a colon-separated list of numbers from `0` to `7`. This field is part of the ETS settings.

Table 39. lsportip output (continued)

Attribute	Description
bandwidth_allocation	Displays a list of guaranteed bandwidth allocation percentages for priority groups zero through seven. The value is either blank or a colon-separated numeric list of bandwidth allocation percentages for each priority group, where each percentage is a whole number integer. This field is part of the ETS settings. **Note:** The field is blank if no specific bandwidths are allocated to any priority group on the network.
lossless_iscsi	Displays whether PFC is enabled (`on`) or not enabled (`off`) for an iSCSI Internet Protocol Version 4 (IPv4) IP address. To be `on`: • It must be a 10 Gbps port with a valid IPv4 address. • PFC for iSCSI is enabled on the switch port. • Virtual local area network (VLAN) is configured for this IPv4 address. • iSCSI host attach is enabled on the port. Otherwise the value is `off`.
lossless_iscsi6	Displays whether PFC is enabled (`on`) or not enabled (`off`) for an iSCSI Internet Protocol Version 6 (IPv6) IP address. To be `on`: • It must be a 10 Gbps port with a valid IPv6 address • PFC for iSCSI is enabled on the switch port • Virtual local area network (VLAN) is configured for this IPv4 address • iSCSI host attach is enabled on the port Otherwise the value is `off`.
storage	Indicates whether the IPv4 address that is used for iSCSI backend storage attach functions. The values are `yes` and `no` (default). If no address is specified, the value is blank.
storage_6	Indicates whether the IPv6 address on the port is used for iSCSI backend storage attach functions. The values are `yes` and `no` (default). If no address is specified, the value is blank.
host_port_grp_id	Indicates the host port group ID in both concise and detailed views. Values are 0 and 1-32. The default value is 0 for any Ethernet port. All configured host attached ports have a non-zero host_port_grp_id. • 0 (default): For a new installation, all configured iSCSI ports that have the host flag set to no have this field set to 0. Upon an update from older versions, all previously configured iSCSI ports are added to the default host port group that is 0. After the update to the current version, even though the host flag is set to yes, iSCSI ports are placed in host port group 0. • 1-32: These host port group IDs are assigned to each configured iSCSI port that has its host flag set to yes. Host port grouping groups the ports that have the same speed and ensures that no more than four ports are discovered by a host during a discovery. A maximum of four ports per system node can belong to the same host port group ID. All ports that belong to the same host port group ID are the same speed. Across two nodes of an I/O group, up to eight iSCSI ports (four per node) can belong to the same host port group ID.

In the following examples, (which list different port configuration options), there are two lines for each possible Ethernet port, which represent the port and iSCSI behavioral effects. Port indices are assigned statically, and higher indices are used for optional ports.

A concise invocation example

```
lsportip -delim ,
```

The resulting output:

```
id,node_id,node_name,IP_address,mask,gateway,IP_address_6,prefix_6,gateway_6,MAC,duplex,state,speed,failover,
link_state,host,remote_copy,host_6,remote_copy_6,remote_copy_status,remote_copy_status_6,vlan,vlan_6,
adapter_location,adapter_port_id,lossless_iscsi,lossless_iscsi6,storage,storage_6,host_port_grp_id
1,1,node1,192.168.48.135,255.255.255.0,192.168.48.1,,,,5c:f3:fc:f5:67:ca,Full,configured,1Gb/s,no,
active,yes,1,,0,unused,,65,,0,1,off,,no,1
1,1,node1,,,,,,5c:f3:fc:f5:67:ca,Full,configured,1Gb/s,yes,active,,0,,0,,,,,0,1,,,,0
2,1,node1,192.168.48.136,255.255.255.0,192.168.48.1,,,,5c:f3:fc:f5:67:cb,Full,configured,1Gb/s,no,
active,yes,1,,0,unused,,,,0,2,off,,no,1
2,1,node1,,,,,,5c:f3:fc:f5:67:cb,Full,configured,1Gb/s,yes,active,,0,,0,,,,,0,2,,,,0
3,1,node1,192.168.48.137,255.255.255.0,192.168.48.1,,,,00:90:fa:27:ec:22,,configured,10Gb/s,no,
active,yes,1,,0,unused,,,,1,1,off,,no,1
3,1,node1,,,,,,00:90:fa:27:ec:22,,configured,10Gb/s,yes,active,,0,,0,,,,,1,1,,,,0
4,1,node1,192.168.48.138,255.255.255.0,192.168.48.1,0009:2009:0003:0004:0005:0006:0007:1130,64,
fe80:0000:0000:0000:0007:b4ff:fe00:0a00,00:90:fa:27:ec:24,,configured,10Gb/s,no,
active,yes,1,yes,0,unused,,165,170,1,2,on,on,yes,yes,1
4,1,node1,,,,,,00:90:fa:27:ec:24,,configured,10Gb/s,yes,active,,0,,0,,,,,1,2,,,,0
1,2,node2,192.168.48.145,255.255.255.0,192.168.48.1,,,,5c:f3:fc:f5:68:b2,Full,configured,1Gb/s,no,
active,yes,1,,0,unused,,65,,0,1,off,,no,1
1,2,node2,,,,,,5c:f3:fc:f5:68:b2,Full,configured,1Gb/s,yes,active,,0,,0,,,,,0,1,,,,0
2,2,node2,192.168.48.146,255.255.255.0,192.168.48.1,,,,5c:f3:fc:f5:68:b3,Full,configured,1Gb/s,no,
active,yes,1,,0,unused,,,,0,2,off,,,1
2,2,node2,,,,,,5c:f3:fc:f5:68:b3,Full,configured,1Gb/s,yes,active,,0,,0,,,,,0,2,,,,0
3,2,node2,192.168.48.147,255.255.255.0,192.168.48.1,,,,00:90:fa:27:ec:4a,,configured,10Gb/s,no,
active,yes,1,,0,unused,,,,1,1,off,,no,1
3,2,node2,,,,,,00:90:fa:27:ec:4a,,configured,10Gb/s,yes,inactive,,0,,0,,,,,1,1,,0
4,2,node2,192.168.48.148,255.255.255.0,192.168.48.1,0009:2009:0003:0004:0005:0006:0007:1230,64,
fe80:0000:0000:0000:0007:b4ff:fe00:0a00,00:90:fa:27:ec:4c,,configured,10Gb/s,no,active,yes,1,yes,
0,unused,,165,170,1,2,on,on,yes,yes,1
4,2,node2,,,,,,00:90:fa:27:ec:4c,,configured,,yes,inactive,,0,,0,,,,,1,2,,,,0
```

A concise invocation example

```
lsportip
```

The resulting output:

```
id node_id node_name IP_address   mask          gateway     IP_address_6                            prefix_6 gateway_6                               MAC               duplex state
1  1       node1     192.168.1.52 255.255.255.0 192.168.1.1                                                                                          5c:f3:fc:0b:da:64 Full   configu
1  1       node1                                                                                                                                     5c:f3:fc:0b:da:64 Full   configu
2  1       node1                                            fc00:0000:0000:0000:445a:0a17:fcf7:0236 64       fc00:0000:0000:0000:445a:0a17:fcf7:0001 5c:f3:fc:0b:da:66 Full   configu
2  1       node1                                                                                                                                     5c:f3:fc:0b:da:66 Full   configu
1  2       node2     192.168.1.53 255.255.255.0 192.168.1.1                                                                                          e4:1f:13:2f:b4:a4 Full   configu
1  2       node2                                                                                                                                     e4:1f:13:2f:b4:a4 Full   configu
2  2       node2                                            fc00:0000:0000:0000:445a:0a17:fcf7:0237 64       fc00:0000:0000:0000:445a:0a17:fcf7:0001 e4:1f:13:2f:b4:a6 Full   configu
2  2       node2                                                                                                                                     e4:1f:13:2f:b4:a6 Full   configu
```

A detailed invocation example

```
lsportip 1
```

The detailed resulting output:

```
id 1
node_id 1
node_name node1
IP_address 192.168.20.10
mask 255.255.255.0
gateway 192.168.20.1
IP_address_6
prefix_6
gateway_6
MAC 00:1a:64:97:1b:a0
duplex Full
state online
speed 1Gb/s
failover no
mtu 1500
host yes
remote_copy 0
host_6
remote_copy_6 0
remote_copy_status
remote_copy_status_6
vlan 1063
vlan_6
```

```
adapter_location 1
adapter_port_id 1
dcbx_state Enabled
iscsi_priority_tag  4
fcoe_priority_tag  3
pfc_enabled_tags 3:4
pfc_disabled_tags  0:1:2:5:6:7
priority_group_0
priority_group_1
priority_group_2
priority_group_3
priority_group_4
priority_group_5
priority_group_6 3
priority_group_7 4
bandwidth_allocation 0:0:0:0:0:0:30:30
lossless_iscsi  on
lossless_iscsi6
storage yes
storage_6
host_port_grp_id 1

id 1
node_id 1
node_name node1
IP_address
mask
gateway
IP_address_6
prefix_6
gateway_6
MAC 00:1a:64:97:1b:a0
duplex Full
state online
speed 1Gb/s
failover yes
mtu 1500
host yes
remote_copy 0
host_6
remote_copy_6 0
remote_copy_status
remote_copy_status_6
vlan 1063
vlan_6
adapter_location 1
adapter_port_id 1
dcbx_state Enabled
iscsi_priority_tag  4
fcoe_priority_tag  3
pfc_enabled_tags 3:4
pfc_disabled_tags  0:1:2:5:6:7
priority_group_0
priority_group_1
priority_group_2
priority_group_3
priority_group_4
priority_group_5
priority_group_6 3
priority_group_7 4
bandwidth_allocation 0:0:0:0:0:0:30:30
lossless_iscsi  on
lossless_iscsi6
storage
storage_6
host_port_grp_id 1
```

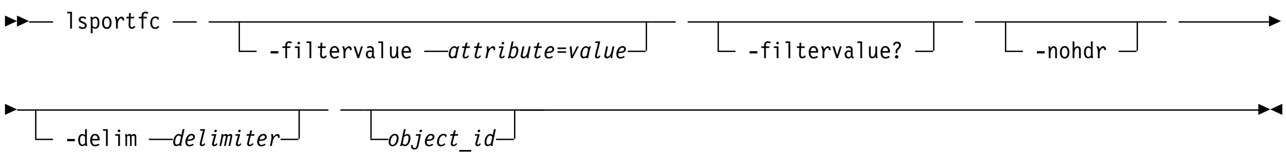
```
id 1
node_id 2
node_name node2
IP_address 192.168.20.11
mask 255.255.255.0
gateway 192.168.20.1
IP_address_6
prefix_6
gateway_6
MAC 00:1a:64:97:16:08
duplex Full
state online
speed 1Gb/s
failover no
mtu 1500
host yes
remote_copy 0
host_6
remote_copy_6 0
remote_copy_status
remote_copy_status_6
vlan 1063
vlan_6
adapter_location 1
adapter_port_id 1
dcbx_state Enabled
iscsi_priority_tag  4
fcoe_priority_tag  3
pfc_enabled_tags 3:4
pfc_disabled_tags  0:1:2:5:6:7
priority_group_0
priority_group_1
priority_group_2
priority_group_3
priority_group_4
priority_group_5
priority_group_6 3
priority_group_7 4
bandwidth_allocation 0:0:0:0:0:0:30:30
lossless_iscsi  on
lossless_iscsi6
storage yes
storage_6
host_port_grp_id 1

id 1
node_id 2
node_name node2
IP_address
mask
gateway
IP_address_6
prefix_6
gateway_6
MAC 00:1a:64:97:16:08
duplex Full
state online
speed 1Gb/s
failover yes
mtu 1500
host yes
remote_copy 0
host_6
remote_copy_6 0
remote_copy_status
remote_copy_status_6
vlan 1063
```

```
vlan_6
adapter_location 1
adapter_port_id 1
dcbx_state Enabled
iscsi_priority_tag  4
fcoe_priority_tag  3
pfc_enabled_tags 3:4
pfc_disabled_tags  0:1:2:5:6:7
priority_group_0
priority_group_1
priority_group_2
priority_group_3
priority_group_4
priority_group_5
priority_group_6 3
priority_group_7 4
bandwidth_allocation 0:0:0:0:0:0:30:30
lossless_iscsi  on
lossless_iscsi6
storage
storage_6
host_port_grp_id 1
```

lsportfc

Use the **lsportfc** command to view the status and properties of the Fibre Channel (FC) input/output (I/O) ports for the clustered system.

Syntax



Parameters

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are returned. If a capacity is specified, the units must also be included.

-filtervalue?
(Optional) Displays the valid filter attributes. The following filter attributes for the **lsportfc** command are valid:

- type
- status
- node_id
- fc_io_port_id
- attachment

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*

(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the -delim parameter is a 1-byte character. If you enter **-delim :** on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

object_id

(Optional) Specifies the ID of an object that is displayed in the view.

Description

This command enables you to view information about clustered system port status.

Table 40 provides the attribute values that can be displayed as output view data.

The following table shows the possible outputs:

Table 40. lsportfc output

Attribute	Description
id	Indicates a unique value for the object. The value must be a numeric 0 or greater.
fc_io_port_id	Indicates the FC I/O port ID. The value must be a positive integer.
port_id	Indicates the platform port ID. The value must be a positive integer.
type	Indicates the type of platform port. The value can be either `fc` or `ethernet`.
port_speed	Indicates the I/O port speed. The value is XGb. The value is N/A if the port has never been active. If the port is inactive, it shows the last-known port speed.
node_id	Indicates the ID of the node containing the port. The value must be a positive integer.
node_name	Indicates the name of the node containing the port.
WWPN	Indicates the I/O port worldwide port name (WWPN). The value must be in 16-character hexadecimal format.
nportid	Indicates the most recent NPort ID used by the port. The value must be in 6-character hexadecimal format, and all zeroes if never active.
status	Indicates whether the port is configured to a device of Fibre Channel (FC) port. The values are: • `active` • `inactive_configured` • `inactive_unconfigured`.
switch_WWPN	Indicates the WWPN of the device that was most recently attached to the port. The value must be in 16-character hexadecimal format, or all zeroes if the port has never been active.
vlan_id	Indicates the VLAN ID on which a specific VN port is communicating. The value is up to a 4-character decimal string. The value is `N/A` for ports that are never active. If the port is inactive, the last-known VLAN ID is used.
fcf_MAC	Indicates the MAC address for the switch attached to the VN port. The value is N/A for ports that are never active. The value is a formatted 48-bit MAC address. If the port is inactive, the last known `fcf_MAC` value is used.
attachment	Indicates if the port is attached to an FC switch or directly to an FC host. (Or, if the port is offline, it specifies which one it was attached to when last online.)

Table 40. lsportfc output (continued)

Attribute	Description
cluster_use	Indicates the node's current capability for local or partner cluster communications: • `none` indicates the port cannot be used for any node communication • `local` indicates the port can be used for local clustered system (system) node communication • `partner` indicates the port can be used for partner system node communication • `local_partner` indicates the port can be used for both local and partner system node communication
adapter_location	Indicates the location of the adapter containing the Ethernet port (any number from 0 to 6).
adapter_port_id	Indicates the location of the Ethernet port that is in the adapter (any number from 1 to 4).
fabric_WWN	Indicates the World Wide Name (WWN) value for the connected fabric string. The value must be in 16-character hexadecimal format, or all zeroes if the port has never been active or the port is not connected to a fabric.

A concise invocation example

```
lsportfc
```

The resulting output:

```
id fc_io_port_id port_id type     port_speed node_id node_name WWPN             nportid status                attachment cluster_use   adapter_location  adapter_port_id
0  1             1       fc       8Gb        1       node1     500507680140BADD 0E2411  active                switch     local_partner 1                 1
1  2             2       fc       8Gb        1       node1     500507680130BADD 0E2412  active                switch     local_partner 1                 2
2  3             3       fc       N/A        1       node1     500507680110BADD 000000  inactive_unconfigured none       partner       1                 3
3  4             4       fc       N/A        1       node1     500507680120BADD 000000  inactive_unconfigured none       none          1                 4
4  5             3       ethernet 10Gb       1       node1     500507680150BADD 0E2413  active                switch     local         2                 1
5  6             4       ethernet 10Gb       1       node1     500507680160BADD 0E2414  inactive_configured   switch     local         2                 2
6  1             1       fc       N/A        2       node2     500507680140BADE 000000  inactive_unconfigured none       local_partner 2                 3
7  2             2       fc       N/A        2       node2     500507680130BADE 000000  inactive_unconfigured none       local_partner 2                 4
8  3             3       fc       N/A        2       node2     500507680110BADE 000000  inactive_unconfigured none       partner       3                 1
9  4             4       fc       N/A        2       node2     500507680120BADE 0E2414  active                switch     none          3                 2
10 5             3       ethernet 10Gb       2       node2     500507680150BADE 0E2415  active                switch     local         3                 3
11 6             4       ethernet 10Gb       2       node2     500507680160BADE 0E2416  active                switch     local         3                 4
```

A detailed invocation example

```
lsportfc 10
```

The detailed resulting output:

```
id 10
fc_io_port_id 5
port_id 3
type ethernet
port_speed 10Gb
node_id 6
node_name node3
WWPN 50050768015051E5
nportid 012701
status active
switch_WWPN 202700053346FA3D
fpma 0E:FC:00:01:27:01
vlanid 100
fcf_MAC 00:05:73:C2:CA:B4
cluster_use none
adapter_location 1
adapter_port_id 1
fabric_WWN 202700053346FA3C
```

lsportsas

Use the **`lsportsas`** command to display the status of all SAS ports in the clustered system.

Syntax

```
►►─ lsportsas ─┬──────────────────────────────────┬─ ┬──────────────────┬─ ┬──────────┬─ ──────►
               └─ -filtervalue ─attribute=value─┘   └─ -filtervalue? ─┘   └─ -nohdr ─┘

►─┬───────────────────────┬──────────────────────────────────────────────────────────────────►◄
  └─ -delim ─delimiter─┘
```

Parameters

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are returned. If a capacity is specified, the units must also be included.

-filtervalue?
(Optional) Displays the valid filter attributes. The following filter attributes for the **lsportsas** command are valid:

- node_id
- status
- attachment
- type

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed view. This parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum allowable width for each data item. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. If you enter **-delim :** on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

Description

This command displays information about system port status.

This command output shows all available paths, which are defined by zoning, independent of their usage. This means that the command output includes paths that are not used because of port masking.

Table 41 provides the attribute values that can be displayed as output view data.

Table 41. lsportsas output

Attribute	Description
id	Indicates the line number within the displayed information (numeric string).
port_id	Indicates the ID of the port.
port_speed	Indicates the speed of the I/O port (in XGb). This speed is the fastest local link speed for the SAS port. The value is the last-known port speed if the port is inactive, and N/A if port has is unused and was never active.
node_id	Indicates the ID of the node that contains the port (numeric string).
node_name	Indicates the name of the node that contains the port (alphanumeric string).

Table 41. lsportsas output (continued)

Attribute	Description
WWPN	Indicates the worldwide port name (WWPN) for the I/O port (16-character hexadecimal string).
status	Indicates the status of the port (numeric string). The values can be: • `online` if the port is functioning • `offline` if the port is not functioning • `offline_unconfigured` if not functioning but not configured by the user • `degraded` if one or more ports are not functioning or have a lower speed than others • `excluded` if excluded by the user or system
switch_WWPN	Indicates the WWPN of the switch port if attached to switch (16-character hexadecimal string), or is blank. If the port is offline, the last-known value is displayed.
attachment	Indicates what the port is attached to. The possible values are: • `drive` • `controller` • `switch` • `host` • `enclosure` • `none` If the port is offline, the field shows what was attached when the port was last online.
type	Indicates how the port is configured. This field also shows the devices that can be attached to the SAS port. The possible values are: • `drive` • `enclosure` • `enclosure_controller` • `host_controller` • `none`
adapter_location	Displays the location of the adapter that contains the SAS port (any number in the range 0 - 6).
adapter_port_id	Displays the location of the SAS port that is in the adapter (any number in the range 1 - 4).

An invocation example

```
lsportsas
```

The resulting output:

```
id port_id port_speed node_id node_name WWPN             status               switch_WWPN      attachment type            adapter_location  adapter_port_id
0  1       3Gb        1       node1     500507680140004A offline                                   enclosure  enclosure       0                  1
1  2       6Gb        1       node1     500507680150004A online               5001234567892000 switch     host_controller 0                  2
4  1       3Gb        2       node2     50050768014051E5 online                                    host       host_controller 0                  3
5  2       3Gb        2       node2     50050768015051E5 offline_unconfigured                      none       none            0                  4
```

An invocation example

```
lsportsas
```

The resulting output:

```
id port_id port_speed node_id node_name WWPN             status switch_WWPN      attachment type     adapter_location  adapter_port_id
0  1       12Gb       1       node1     50050768056C009E online 500507680600B63F enclosure  enclosure  0                  0
1  2       12Gb       1       node1     50050768056C009F online 500507680600B64F enclosure  enclosure  0                  1
2  0       12Gb       1       node1     50050768056C009G online 500507680600B65F enclosure  internal  0                  2
4  1       12Gb       2       node2     50050768056C009I online 500507680600B66F enclosure  enclosure  0                  3
5  2       12Gb       2       node2     50050768056C009J online 500507680600B67F enclosure  enclosure  0                  4
6  0       12Gb       2       node2     50050768056C009K online 500507680600B68F enclosure  internal  1                  1
```

lsquorum

Use the **lsquorum** command to list the quorum devices that the clustered system (system) uses to store quorum data.

Syntax

```
►►— lsquorum — ┬─────────┬─ ┬────────────────────┬─ ┬────────────────┬──────────────►◄
               └ -nohdr ┘   └ -delim — delimiter ┘   └ quorum_index ┘
```

Parameters

-nohdr

(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*

(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **delim** parameter overrides this behavior. Valid input for the **delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified character.

quorum_index

(Optional) Specifies the quorum device by its index number. When you use this parameter, a detailed view of the specified device is returned. If you do not specify a device, then a concise view of all quorum devices is displayed.

Description

This command displays a concise list or a detailed view of the MDisks or drives that the system is using to store quorum data. This information can be used to ensure that the quorum candidates are on separate storage subsystems.

Note: The object type is either MDisk or drive, but only MDisks are used to hold quorum data. If the quorum object type is a drive, the controller ID and name fields are blank.

Table 42 provides the attribute values that can be displayed as output view data.

Table 42. lsquorum output

Attribute	Possible Values
quorum_index	Indicates the quorum device by index number.
status	Indicates the quorum device status.
name	Indicates the name of the object that is used as the quorum device.
controller_id	Indicates the ID of the controller of an MDisk object that is used as the quorum device.
controller_name	Indicates the name of the controller of an MDisk object that is used as the quorum device.

Table 42. lsquorum output (continued)

Attribute	Possible Values
active	Indicates whether this quorum is the active quorum device the system uses as a tie breaker.
object_type	Indicates the type of object the quorum device uses.
override	Indicates whether the automatic quorum selection for this quorum device was overridden.
site_id	Indicates the site value for the quorum device. This numeric value is 1, 2, 3, or blank.
site_name	Indicates the site name for the quorum device (MDisks or drives). This is an alphanumeric value or is blank. For the quorum application, this name identifies the site it is deployed at. By default, this name is the local host's IP address, but a custom alias can also be set.

A concise invocation example

```
lsquorum
```

The concise resulting output:

```
quorum_index  status  id  name    controller_id  controller_name  active  object_type  override  site_id  site_name
0             online  1   mdisk1  1              controller1      no      mdisk        no        2        site2
1             online  2   mdisk2  1              controller1      no      mdisk        no        1        site1
2             online                                              yes     device       no                 quorumhost/9.15
```

A detailed invocation example

```
lsquorum 1
```

The detailed resulting output:

```
quorum_index 1
status online
id 309
name mdisk9
controller_id 1
controller_name controller3
active yes
object_type drive
override yes
site_id 1
site_name CPD1
```

```
quorum_index 2
status online
id 33
name
controller_id
controller_name
active no
object_type drive
override no
site_id 1
site_name CPD1
```

lsroute

Use the **lsroute** command to display the IP routing table.

Syntax

```
►►— lsroute — ┬────────────────────────┬─ ┬──────────┬──────────────────────────────►◄
              └─ -delim — delimiter ──┘   └─ -nohdr ─┘
```

Parameters

-nohdr

(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*

(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

Description

This command displays the IP routing table. The table provides details of the gateway that is used for IP traffic to a range of IP addresses for each Ethernet port. This information can be used to diagnose configuration node accessibility problems. The **lsroute** command is equivalent to the Linux **route** command.

An invocation example

```
lsroute
```

The resulting output:

```
Kernel IP routing table
Destination     Gateway       Genmask         Flags Metric Ref    Use Iface
9.71.46.0       0.0.0.0       255.255.254.0   U     0      0        0 eth0
127.0.0.0       0.0.0.0       255.0.0.0       U     0      0        0 lo
0.0.0.0         9.71.46.1     0.0.0.0         UG    0      0        0 eth0
```

```
Kernel IPv6 routing table
Destination                             Next Hop                Flags Metric Ref  Use Iface
2002:914:fc12:849::/64                  ::                      UA    256    3675   0 eth0
fe80::/64                               ::                      U     256    0      0 eth0
::/0                                    fe80::7:b4ff:fe00:500 UGDA  1024   1      0 eth0
::1/128                                 ::                      U     0      1441   1 lo
2002:914:fc12:849:214:5eff:fe33:5192/128 ::                     U     0      0      1 lo
fe80::214:5eff:fe33:5192/128            ::                      U     0      0      1 lo
ff00::/8                                ::                      U     256    0      0 eth0
```

lstimezones

Use the **lstimezones** command to list the time zones that are available on the clustered system (system). Each timezone is assigned an ID that can be used in the **settimezone** command to set the time zone.

Syntax

```
►►── lstimezones ── ┬─────────┬─ ┬─────────────────────┬──────────────────────────►◄
                    └─ -nohdr ─┘  └─ -delim ── delimiter ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (**:**) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by a colon character.

Description

This command displays a list of all the time zones that are available on the system. Each time zone is assigned an ID. This ID can be used in the **settimezone** command.

An invocation example

```
lstimezones
```

The resulting output:

```
id timezone
0 Africa/Abidjan
1 Africa/Accra
2 Africa/Addis_Ababa
3 Africa/Algiers
4 Africa/Asmera
5 Africa/Bamako
6 Africa/Bangui
```

lssasportcandidate

Use the **lssasportcandidate** command to list the unconfigured serial-attached SCSI (SAS) ports that are logged in and available to add to the SAS worldwide port name (WWPN) or host objects.

Syntax

```
►►── lssasportcandidate ── ┬─────────┬─ ┬─────────────────────┬────────────────────►◄
                           └─ -nohdr ─┘  └─ -delim ── delimiter ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

Description

This command returns a list of unconfigured, logged in SAS ports.

Note: The **lssasportcandidate** command presents a list of host SAS ports that are logged in to nodes. However, there are situations when the list might include host SAS ports that are no longer logged in or even part of the SAN fabric. For example, if a host SAS port is unplugged from a switch but **lssasportcandidate** shows the WWPN that is logged in to all nodes, the incorrect entry is removed when another device is plugged in to the same switch port that previously contained the removed host SAS port.

This table shows the possible output:

Table 43. lssasportcandidate output

Attribute	Description
sas_WWPN	Indicates that the SAS WWPN that is logged in is unconfigured (not assigned to a host). This value must be 16 hexadecimal characters.

An invocation example

```
lssasportcandidate
```

The resulting output:

```
sas_WWPN
200600A0B813B7AC
200600A0B813B7AD
```

lssecurity

Use the **lssecurity** command to display the current system Secure Sockets Layer (SSL) or Transport Layer Security (TLS) security settings.

Syntax

```
►►—lssecurity— ┬──────────┬ ┬─────────────────────┬ ————————————————————————►◄
               └ -nohdr ──┘ └ -delim — delimiter ─┘
```

Parameters

-nohdr

(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If no data exists to be displayed, headings are not displayed.

-delim *delimiter*

(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (**:**) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

Description

This command displays the current system SSL, SSH, or TLS security settings.

This table provides the possible values that are displayed for the **lssecurity** command.

Table 44. **lssecurity** *attribute values*

Attribute	**Value**
sslprotocol	Identifies the current security level setting, a numeric value of 1, 2, 3, or 4. A security level setting of: • 1 allows TLS 1.0, TLS 1.1, and TLS 1.2, but disallows SSL 3.0. • 2 disallows TLS 1.0 and TLS 1.1. • 3 also disallows TLS 1.2 cipher suites that are not exclusive to 1.2. 2 • 4 additionally disallows RSA key exchange ciphers. **Note:** You cannot use the management GUI if the sslprotocol value is set to 1 and you are using SSL 3.0 or TLS 1.0.

*Table 44. **lssecurity** attribute values (continued)*

Attribute	Value
sshprotocol	Identifies the current security level for SSH, a numeric value of 1 or 2. A security level setting of: • 1 allows the following key exchange methods. – curve25519-sha256 – curve25519-sha256@libssh.org – ecdh-sha2-nistp256 – ecdh-sha2-nistp384 – ecdh-sha2-nistp521 – diffie-hellman-group-exchange-sha256 – diffie-hellman-group16-sha512 – diffie-hellman-group18-sha512 – diffie-hellman-group14-sha256 – diffie-hellman-group14-sha1 – diffie-hellman-group1-sha1 – diffie-hellman-group-exchange-sha1 • 2 allows the following key exchange methods. – curve25519-sha256 – curve25519-sha256@libssh.org – ecdh-sha2-nistp256 – ecdh-sha2-nistp384 – ecdh-sha2-nistp521 – diffie-hellman-group-exchange-sha256 – diffie-hellman-group16-sha512 – diffie-hellman-group18-sha512 – diffie-hellman-group14-sha256 – diffie-hellman-group14-sha1

An invocation example

```
lssecurity
```

The resulting output:

```
sslprotocol 4
sshprotocol 1
```

lssite

Use the **lssite** command to report the names of the sites.

Syntax

```
►►—lssite— ─┬─────────┬─ ─┬──────────────────────┬─ ────────────────────────────►◄
            └─ -nohdr ─┘   └─ -delim — delimiter ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

Description

This command reports the names of the sites.

Remember: This command is only applicable when a system is configured as a stretched system or a HyperSwap system (specify the **chsystem -topology** command).
In a stretched configuration these applications are spread across two or more geographic locations or sites:

- Nodes
- Storage
- Host servers
- Infrastructure

Table 45 provides the attribute values that can be displayed as output view data.

Table 45. lssite attribute values

Attribute	Value
id	Identifies the numeric value that represents the site. The value can be 1, 2, or 3.
name	Identifies the site name.

An invocation example

```
lssite
```

The resulting output:

```
id name
1  CPD1
2  CPD2
3  Quorum
```

lssra

Use the **lssra** command to check both secure remote assistance status and the time of the last login.

Syntax

```
►►── lssra ── ┬────────┬─ ┬──────────────┬─ ┬─────────────────────────────┬─ ── ───►
              └─ -nohdr ┘  └─ -filtervalue? ┘  └─ -filtervalue ── attribute=value ┘

►┬──────────────────┬─ ─────────────────────────────────────────────────────►◄
 └─ -delim ── delimiter ┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If no data needs to be displayed, headings are not displayed.

-filtervalue?
(Optional) Displays a list of valid filter attributes for the **-filtervalue***attribute=value* parameter. The valid filters for the **lssra** command are:

- port_id
- owning_node_id
- current_node_id
- host_io_permitted
- virtualized

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are returned. If a capacity is specified, the units must also be included.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards when you are using the CLI:

- The wildcard character is an asterisk (*).
- The command can contain a maximum of one wildcard.
- When you are using a wildcard, you must enclose the filter entry within double quotation marks ("").

-config_node
(Optional) Specifies the local field values for the current configuration node. Do not specify this parameter if the specified node is part of an active clustered system. This parameter is mutually exclusive with all other parameters.

Description

This command displays the both secure remote assistance status and the time of the last login.

Table 46 provides the attribute values that can be displayed as output view data.

Table 46. lssra output

Attribute	**Description**
status	Indicates whether the support assistance is enabled or not enabled (default).
token_age_in_days	Displays the number of days that the current token exists.
active_monitor_user_count	Displays the number of support assistance monitor users that are logged in to this system.
monitor_user_last_login	Displays the time of the last monitor user login in the format YYMMDDHHMMSS. The default is blank.

Table 46. lssra output (continued)

Attribute	**Description**
active_privileged_user_count	Displays the number of secure remote access privileged users that are currently logged on.
privileged_user_last_login	Displays the date and time of the last support assistance privileged user login on the system in the format `YYMMDDHHMMSS`. The default is blank.
remote_support_test_status	Indicates the remote system support service state when it was last tested. The values are: • `disconnected` (default) • `connecting` • `connected` • `active` • `failure` This value is the maximum state that is reached on the configuration node, in increasing order from `disconnected` through to `failure`.
remote_support_test_time	Indicates the time stamp of the last remote system support test. The value must be in the format `YYMMDDHHMMSS`.
remote_support_config_changed_after_test	Indicates that a support center was added since the system was last tested. The values are `yes` or `no` (default).
remote_support_enabled	Indicates whether remote support is enabled. The values are `yes` or `no` (default).
remote_support_status	Indicates the remote support service status or state. If you do not enable remote support the status is `disconnected`. The values are: • `disconnected` (default) • `connecting` • `connected` • `active` • `failure` This value is the maximum state across all online nodes, in increasing order from `disconnected` through to `failure`.
remote_support_enabled_time	Indicates the time stamp for the last successful creation of a secure tunnel by a remote support service. The value must be in the format `YYMMDDHHMMSS`.
remote_support_idletimeout	Indicates the idle timeout value. The value must be a number (non-negative), and the default value is `0`.
remote_support_center_id	Indicates the support center ID (specified by using **`lssystemsupportcenter`**that is used to establish the secure tunnel. The value must be a number 0 -11, and the default is blank.

An invocation example on a system with local support assistance enabled

```
lssra
```

The detailed resulting output:

```
status enabled
active_monitor_user_count 0
monitor_user_last_login
active_privileged_user_count 0
privileged_user_last_login
token_age_in_days 7
remote_support_test_status connected
remote_support_test_time 161123183137
remote_support_config_changed_after_test no
```

```
remote_support_enabled no
remote_support_status disconnected
remote_support_enabled_time
remote_support_idletimeout 0
remote_support_center_id
```

An invocation example on a system with remote support assistance enabled

```
lssra
```

The detailed resulting output:

```
status enabled
active_monitor_user_count 0
monitor_user_last_login
active_privileged_user_count 1
privileged_user_last_login   161123204006
token_age_in_days 30
remote_support_test_status connected
remote_support_test_time 1611231530220
remote_support_config_changed_after_test no
remote_support_enabled yes
remote_support_status active
remote_support_enabled_time 161123183137
remote_support_idletimeout 0
remote_support_center_id 0
```

lsthrottle

Use the **lsthrottle** command to list throttle objects that are configured in the clustered system.

Syntax

```
►►── lsthrottle ── ┬─────────┬─ ┬────────────────┬─ ┬────────────────────────────────┬──►
                   └─ -nohdr ┘  └─ -filtervalue? ┘  └─ -filtervalue ── attribute=value ┘

►─ ┬──────────────────────┬─ ──────────────────────────────────────────────────────►◄
   └─ -delim ── delimiter ┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If no data is displayed, headings are not displayed.

-filtervalue?
Displays a list of valid filter attributes for the **-filtervalue***attribute=value* parameter. The valid filters for the **lsthrottle** command are:

- throttle_type

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are returned. If a capacity is specified, the units must also be included.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards when you are using the CLI:

- The wildcard character is an asterisk (*).

- The command can contain a maximum of one wildcard.
- When you are using a wildcard, you must enclose the filter entry within double quotation marks ("").

-delim *delimiter*

(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

Description

This command lists throttle objects that are configured in the clustered system.

Table 47 provides the attribute values that can be displayed as output view data.

Table 47. lsthrottle output

Attribute	Description
throttle_id	Indicates the unique ID for the throttle object. The value is a number 0 - 10144.
throttle_name	Indicates the unique name for the throttle object. The value is an alphanumeric string that is 63 characters long.
object_id	Indicates the ID of the object to which throttle is applied. The value is a number 0 - 8191.
object_name	Indicates the name of the object to which throttle is applied. The value is an alphanumeric string that is 63 characters long.
throttle_type	Indicates the type of throttle object. The values are: `offload`, `vdisk`, `host`, `hostcluster`, and `mdiskgrp`.
IOPs_limit	Indicates the limit of configured IOPs. The value is a numeric string in the range 0 - 33554432. If no limit is specified the value is blank.
bandwidth_limit_MB	Indicates the bandwidth (in MBps). The value is a numeric string in the range 0 - 268435456. If no limit is specified the value is blank.

An invocation example

```
lsthrottle
```

The detailed resulting output:

```
throttle_id throttle_name object_id object_name   throttle_type IOPs_limit bandwidth_limit_MB
0           throttle0     1         R48U20_213    host                     40
1           throttle1     0         WinHostClust  hostcluster  8000
2           throttle2     9         vdisk0        vdisk                    20
3           throttle3     11        mdiskgrp0     mdiskgrp                 100
0           throttle4                             offload                  500
```

lssystem

Use the **lssystem** command to display a detailed view of a clustered system (system).

Syntax

```
►►— lssystem — ┬──────────┬ ┬─────────────────────┬──────────────────────────────►◄
               └─ -nohdr ─┘ └─ -delim — delimiter ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If no data exists to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a character with 1 byte. In a detailed view, the data is separated from its header by the specified delimiter.

Description

This command displays a detailed view of a system.

Table 48 provides the attribute values that can be displayed as output view data.

Table 48. **`lssystem`** *output*

Attribute	Possible Values
layer	The value can be either: • `replication`, which means the system can create partnerships. • `storage` (default), which means the system can present storage.
location	The location is `local` or `remote`.
statistics status	The status is `on` or `off`.
auth_service_type	Native Lightweight Directory Access Protocol (LDAP)
auth_service_configured	True if the `auth_service_type` is configured to be LDAP-only (if at least one LDAP server is configured).
auth_service_enabled	True if the `auth_service_type` is configured.
email_state	The possible values are: • `running` • `stopped` • `invalid`

*Table 48. **lssystem** output (continued)*

Attribute	**Possible Values**
partnership	The possible values are: • `fully_configured` • `partially_configured_local` • `partially_configured_local_stopped` • `not_present` • `fully_configured_stopped` • `fully_configured_remote_stopped` • `fully_configured_local_excluded` • `fully_configured_remote_excluded` • `fully_configured_exceeded` • Blank
tier	Indicates which system information is being reported. The values are: • `tier0_flash` • `tier1_flash` • `tier_enterprise` • `tier_nearline`
tier_capacity	Indicates the total MDisk storage in the tier.
tier_free_capacity	Indicates the amount of MDisk storage in the tier that is unused.
compression_active	Indicates whether any compressed volume copies exist in non-data reduction pools. Compressed volumes that are in data reduction pools do not count towards this value.
compression_virtual_capacity	Indicates the total virtual capacity for all compressed volume copies in non-data reduction pools. Compressed volumes that are in data reduction pools do not count towards this value. This value is in unsigned decimal format.
compression_compressed_capacity	Indicates the total used capacity for all compressed volume copies in non-data reduction pools. Compressed volumes that are in data reduction pools do not count towards this value. This value is in unsigned decimal format.
compression_uncompressed_capacity	Indicates the total uncompressed used capacity for all compressed volume copies in non-data reduction pools. Compressed volumes that are in data reduction pools do not count towards this value. This value is in unsigned decimal format.
physical_capacity	Indicates the total physical capacity of all provisioning groups that provide MDisks virtualized by the system. The value must be a number (indicated in units) that is rounded to two decimal places.
physical_free_capacity	Indicates the total free physical capacity of all provisioning groups that provide MDisks virtualized by the system. The value must be a number (indicated in units) that is rounded to two decimal places.

Table 48. **lssystem** *output (continued)*

Attribute	**Possible Values**
total_reclaimable_capacity	Indicates the unused (free) capacity that will be available after data is reduced. For storage pools that are not data reduction pools, this field reports `0.00MB`. **Note: `total_reclaimable_capacity`** is a sum of all the **`lsmdiskgrp reclaimable_capacities`**, which has a +/-1% margin. Therefore, this field has up to a +/-4% margin.
used_capacity_before_reduction	Indicates the total amount of data that is written to thin-provisioned and compressed volume copies that are in data reduction storage pools - before data reduction occurs. This value does not include fully allocated volumes (that can be created in a data reduction storage pool) because they are not eligible for reduction.
used_capacity_after_reduction	Indicates the total amount of capacity that is used for thin-provisioned and compressed volume copies in the storage pool after data reduction occurs.
overhead_capacity	Indicates the overhead capacity consumption in all storage pools that is not attributed to data.
rc_buffer_size	Indicates the resource buffer size that is assigned for Metro Mirror, Global Mirror, and HyperSwap Copy Services.
has_nas_key	The value is `yes` or `no`.
total_drive_raw_capacity	The total known capacity of all discovered drives (regardless of drive use).
email_organization	Indicates the user's organization that is shown in the call home email function.
email_machine_address	Indicates the user's mailing address that is shown in the call home email function.
email_machine_city	Indicates the user's city that is shown in the call home email function.
email_machine_state	Indicates the user's state that is shown in the call home email function.
email_machine_zip	Indicates the user's postal code that is shown in the call home email function.
email_machine_country	Indicates the user's country that is shown in the call home email function.
cache_prefetch	Indicates whether cache prefetching is enabled across the system. The values are `on` and `off`.
local_fc_port_mask	Indicates the Fibre Channel (FC) input/output (I/O) ports that a system can use for node-to-node communications on a local system if those FC I/O ports exist on a node. The value is 64 binary bits.
partner_fc_port_mask	Indicates the FC I/O ports that a system can use for system-to-system communications on a partner system if those FC I/O ports exist on a node. The value is 64 binary bits.
topology	Indicates the system topology: • `standard` • `stretched` • `hyperswap`

Table 48. **lssystem** *output (continued)*

Attribute	Possible Values
topology_status	Indicates the system topology status: - `dual_site` - `recovered_site_1` - `recovered_site_2`
compression_destage_mode	Indicates the cache destage mode operation for Real-time compression within the cluster.
rc_auth_status	Indicates remote copy authentication. Remote copy includes Metro Mirror, Global Mirror, and HyperSwap. The values are: - `chap` - `none` (default)
vdisk_protection_time	Indicates the volume protection time in minutes (whether volume protection is enabled or not enabled). The value must be a number from 15 (default) to 1440.
vdisk_protection_enabled	Indicates whether volume protection is enabled (`yes`) or disabled (`no`) for a system.
product_name	Indicates the product name (an alphanumeric string of no more than 62 characters).
odx	Indicates whether offloaded data transfer (ODX) is enabled or disabled. The values are `on` and `off`.
easy_tier_acceleration	Indicates Easy Tier and pool balance acceleration status. The values are `on` and `off`.
max_replication_delay	Indicates the value for maximum replication delay and is a numeric value in the range 0 - 360.
partnership_exclusion_threshold	Indicates the partnership exclusion threshold value and is a numeric value in the range 30 - 315.
ibmcustomer	Indicates the customer number. The value is blank or is a number that contains 7 - 10 digits.
ibmcomponent	Indicates the component. The value is blank or `SANVCNSW1`.
ibmcountry	Indicates the country. The value is blank or a 3-digit number.
tier_0_flash_compressed_data_used	Indicates the capacity of compressed data that is used on the `flash` tier 0 storage tier. The value must be a number with two decimal places.
tier_1_flash_compressed_data_used	Indicates the capacity of compressed data that is used on the `flash` tier 1 storage tier. The value must be a number with two decimal places.
tier_enterprise_compressed_data_used	Indicates the capacity of compressed data that is used on the tier 2 `enterprise` storage tier. The value must be a number with two decimal places.
tier_nearline_compressed_data_used	Indicates the capacity of compressed data that is used on the tier 3 `nearline` storage tier. The value must be a number with two decimal places.

Table 48. **lssystem** *output (continued)*

Attribute	Possible Values
enhanced_callhome	Indicates whether to collect enhanced data in the call home report. The values are `on` or `off`. The enhanced reports include operational and event-related data and specific configuration information that is included in the inventory report. This function alerts the support center about hardware failures and potentially serious configuration or environmental issues. The support center can use the configuration information to automatically generate best practices or recommendations that are based on your actual configuration.
censor_callhome	Indicates that sensitive data is stripped from enhanced call home data. The values are `on` or `off`.
unmap	Indicates whether the system administrator enabled the Small Computer System Interface (SCSI) unmap feature. The values are `on` (default) or `off`.
total_mdisk_capacity	Indicates the sum of **mdiskgrp** `capacity` plus the capacity of all unmanaged MDisks.
space_in_mdisk_grps	Indicates the sum of **mdiskgrp** `capacity`.
space_allocated_to_vdisks	Indicates the sum of **mdiskgrp** `real_capacity`.
total_free_space	Indicates the sum of **mdiskgrp** `free_capacity`.
total_vdiskcopy_capacity	Indicates the total virtual capacity of all volume copies in the cluster.
total_used_capacity	Indicates the sum of **mdiskgrp** `used_capacity`.
total_vdisk_capacity	Indicates the total virtual capacity of volumes in the cluster.
total_allocated_extent_capacity	Indicates the total size of all extents that are allocated to VDisks or otherwise in use by the system.
total_overallocation	Indicates the `total_vdiskcopy_capacity` as a percentage of `total_mdisk_capacity`. If `total_mdisk_capacity` is zero, then `total_overallocation` should display `100`.
tier0_flash_compressed_data_used	Indicates the capacity of compressed data used on the flash tier 0 storage tier.
tier1_flash_compressed_data_used	Indicates the capacity of compressed data used on the flash tier 1 storage tier.
deduplication_capacity_saving	Indicates the total amount of used capacity saved by data deduplication. This saving is before any compression.
compression_opportunity	Indicates the total amount of capacity of all volume-copies within data reduction pools that are enabled for compression. It does not include Real-time compression capacity or capacity that has been saved by data deduplication.
deduplication_opportunity	Indicates the total amount of `used_capacity_before_reduction` of all volume-copies within data reduction pools that are enabled for data deduplication.

Information about the remote system is reported by the **lssystem** command if you issue either the **mkfcpartnership** or **mkippartnership** command from the local system to the remote system. For example, if the partnership is at least partially established from the local system.

Issue the **lssystem** command to display a detailed view of the system.

Detailed view shows the fields that are described for remote systems only; if the system **location** is **local**, then **partnership** and **bandwidth** do not apply (and are not defined or provided). For a remote system, these fields indicate the following information:

location
: **remote** or **local**

partnership

fully_configured
: The **mkfcpartnership** or **mkippartnership** command is issued in both directions and the remote system is online and available.

partially_configured_local
: The **mkfcpartnership** or **mkippartnership** command is issued from the local system to the remote system. The remote system is online and available for partnership.

partially_configured_local_stopped
: The **mkfcpartnership** or **mkippartnership** command is issued from the local system to the remote system. The **chpartnership** command with the **stop** parameter is issued from the local system, and the remote system is online and available. Issue the **chpartnership** command with the **start** parameter on the local system, and **mkfcpartnership** or **mkippartnership** on the remote system.

not_present
: The **mkfcpartnership** or **mkippartnership** command is issued from the local system to the remote system, and the remote system is not available. Either the remote system is offline, or it is not connected to the local system.

fully_configured_stopped
: The **mkfcpartnership** or **mkippartnership** command is issued in both directions and the remote system is online and available. The **chpartnership** command with the **stop** parameter is issued from the local system.

fully_configured_remote_stopped
: The **mkfcpartnership** or **mkippartnership** command is issued in both directions and the remote system is online and available. The **chpartnership** command with the **stop** parameter is issued from the remote system.

fully_configured_local_excluded
: The **mkfcpartnership** or **mkippartnership** command is issued in both directions. The local system is excluding the connection to the remote system due to too many problems, or either system in the partnership is unable to sustain the I/O workload for the Metro Mirror, Global Mirror, or HyperSwap relationships.

fully_configured_remote_excluded
: The **mkfcpartnership** or **mkippartnership** command has been issued in both directions. The remote system is excluding the connection to the local system due to too many problems, or either system in the partnership is unable to sustain the I/O workload for the Metro Mirror, Global Mirror, or HyperSwap relationships.

fully_configured_exceeded
: Too many systems exist in the system network, and the partnership from the local system to the remote is disabled. Refer to the 1710 or 1720 errors in the system error log at the local and remote system.

bandwidth

The bandwidth available on the intersystem link for background copy, in megabytes per second (MBps).

Important: For partnerships over IP links with compression, this parameter specifies the aggregate bandwidth after the compression had been applied to the data. Do not set this parameter higher than the physical link bandwidth multiplied by the (carefully rounded down) compression factor.

The `console_IP` field displays either the:

- Automatically populated system port 1 IP Address - Internet Protocol Version 4 (IPv4) or IPv6
- User-populated IPv4 address

The port value is always *443*, which requires the system to run by using default Hypertext Transfer Protocol Secure (HTTPS).

A concise invocation example

```
lssystem delim :
```

The resulting output:

```
id:name:location:partnership:id_alias
000002006420A162:system0:local::000002006420A162
```

A detailed invocation example

```
lssystem -delim :
```

The resulting output:

```
id:00000200A2600906
name:tbcluster-29
location:local
partnership
bandwidth
total_mdisk_capacity:60.5TB
space_in_mdisk_grps:60.5TB
space_allocated_to_vdisks:643.74GB
total_free_space:59.9TB
total_vdiskcopy_capacity:663.46GB
total_used_capacity:560.99GB
total_overallocation:1
total_vdisk_capacity:501.25GB
total_allocated_extent_capacity:792.50GB
statistics_status:on
statistics_frequency:15
cluster_locale:en_US
time_zone:375 Europe/London
code_level:6.4.0.0 (build 64.6.1205081000)
console_IP:9.71.53.69:443
id_alias:00000200A2600906
gm_link_tolerance:300
gm_inter_cluster_delay_simulation:0
gm_intra_cluster_delay_simulation:0
gm_max_host_delay:5
email_reply
email_contact
email_contact_primary
email_contact_alternate
email_contact_location
email_contact2
email_contact2_primary
email_contact2_alternate
email_state stopped
```

```
inventory_mail_interval:0


iscsi_auth_method:chap
iscsi_chap_secret:MYCLUSTERCHAP
auth_service_configured:no
auth_service_enabled:no
auth_service_url
auth_service_user_name
auth_service_pwd_set:no
auth_service_cert_set:no
auth_service_type:ldap
relationship_bandwidth_limit:25


tier:tier0_flash
tier_capacity:1.63TB
tier_free_capacity:1.63TB
tier:tier1_flash
tier_capacity:1.63TB
tier_free_capacity:1.63TB
tier:tier_enterprise
tier_capacity:0.00MB
tier_free_capacity:0.00MB
tier:tier_nearline
tier_capacity:0.00MB
tier_free_capacity:0.00MB
has_nas_key:no
layer:replication
rc_auth_method:none
rc_buffer_size:48
compression_active:yes
compression_virtual_capacity:1000.00MB
compression_compressed_capacity:0.41MB
compression_uncompressed_capacity:512.05MB
data_reduction:yes
total_reclaimable_capacity:10.00MB
used_capacity_before_reduction:10.00MB
used_capacity_after_reduction:20.00MB
overhead_capacity:50.00MB
cache_prefetch:on
email_organization:UEFA
email_machine_address:1 Chelsea Blvd
email_machine_city:Fulham
email_machine_state:XX
email_machine_zip:OU812
email_machine_country:GB
total_drive_raw_capacity:900.00GB
easy_tier_acceleration off
local_fc_port_mask:0000000000000000000000000000000000000000000000000000000000001101
partner_fc_port_mask:0000000000000000000000000000000000000000000000000000000000000011

topology:standard
topology_status:
compression_destage_mode:false
vdisk_protection_time:120
vdisk_protection_enabled no
product_name:Compumarge
odx:on
max_replication_delay:222
partnership_exclusion_threshold:120

ibmcustomer:1234567
ibmcomponent:SANVCNSW1
ibmcountry:866
tier_0_flash_compressed_data_used:0.00MB
tier_1_flash_compressed_data_used:0.00MB
```

```
tier_enterprise_compressed_data_used:0.00MB
tier_nearline_compressed_data_used:0.00MB
physical_capacity:0.00MB
physical_free_capacity:0.00MB
enhanced_callhome:on
censor_callhome:off
unmap:off
deduplication_capacity_saving:100GB
overhead_capacity:23.00GB
```

A detailed invocation example

```
lssystem -delim :
```

The resulting output:

```
id:00000200A2600906
name:tbcluster-29
location:local
partnership
bandwidth
total_mdisk_capacity:60.5TB
space_in_mdisk_grps:60.5TB
space_allocated_to_vdisks:643.74GB
total_free_space:59.9TB
total_vdiskcopy_capacity:663.46GB
total_used_capacity:560.99GB
total_overallocation:1
total_vdisk_capacity:501.25GB
total_allocated_extent_capacity:792.50GB
statistics_status:on
statistics_frequency:15
cluster_locale:en_US
time_zone:375 Europe/London
code_level:6.4.0.0 (build 64.6.1205081000)
console_IP:9.71.53.69:443
id_alias:00000200A2600906
gm_link_tolerance:300
gm_inter_cluster_delay_simulation:0
gm_intra_cluster_delay_simulation:0
gm_max_host_delay:5
email_reply
email_contact
email_contact_primary
email_contact_alternate
email_contact_location
email_contact2
email_contact2_primary
email_contact2_alternate
email_state stopped
inventory_mail_interval:0


iscsi_auth_method:chap
iscsi_chap_secret:MYCLUSTERCHAP
auth_service_configured:no
auth_service_enabled:no
auth_service_url
auth_service_user_name
auth_service_pwd_set:no
auth_service_cert_set:no
auth_service_type:tip
relationship_bandwidth_limit:25
tier:ssd
tier_capacity:0.00MB
tier_free_capacity:0.00MB
tier:enterprise
tier_capacity:60.49TB
```

```
tier_free_capacity:59.72TB
has_nas_key:no
layer:replication
rc_auth_method:none
rc_buffer_size:48
compression_active:yes
compression_virtual_capacity:1000.00MB
compression_compressed_capacity:0.41MB
compression_uncompressed_capacity:512.05MB
cache_prefetch:on
email_organization:UEFA
email_machine_address:1 Chelsea Blvd
email_machine_city:Fulham
email_machine_state:XX
email_machine_zip:OU812
email_machine_country:GB
total_drive_raw_capacity:900.00GB
easy_tier_acceleration off
local_fc_port_mask:0000000000000000000000000000000000000000000000000000000000001101
partner_fc_port_mask:0000000000000000000000000000000000000000000000000000000000000011

topology:standard
topology_status:
compression_destage_mode false
vdisk_protection_time 120
vdisk_protection_enabled no
product_name Compumarge
odx on
max_replication_delay 222
```

A detailed invocation example

```
lssystem
```

The resulting output:

```
id 000002006C40A278
name cluster0
location local
partnership
bandwidth
total_mdisk_capacity 222.2GB
space_in_mdisk_grps 0
space_allocated_to_vdisks 0.00MB
total_free_space 222.2GB
total_vdiskcopy_capacity 0.00MB
total_used_capacity 0.00MB
total_overallocation 0
total_vdisk_capacity 0.00MB
total_allocated_extent_capacity 0.00MB
statistics_status on
statistics_frequency 15
cluster_locale en_US
time_zone 522 UTC
code_level 6.4.0.0 (build 61.9.1112130001)
console_IP 0.0.0.0:443
id_alias 000002006C40A278
gm_link_tolerance 300
gm_inter_cluster_delay_simulation 0
gm_intra_cluster_delay_simulation 0
gm_max_host_delay 5
email_reply
email_contact
email_contact_primary
email_contact_alternate
email_contact_location
email_contact2
email_contact2_primary
```

```
email_contact2_alternate
email_state stopped
inventory_mail_interval 0
cluster_ntp_IP_address
cluster_isns_IP_address
iscsi_auth_method none
iscsi_chap_secret
auth_service_configured no
auth_service_enabled no
auth_service_url
auth_service_user_name
auth_service_pwd_set no
auth_service_cert_set no
auth_service_type tip
relationship_bandwidth_limit 25

tier tier0_flash
tier_capacity 1.63TB
tier_free_capacity 1.63TB
tier tier1_flash
tier_capacity 1.63TB
tier_free_capacity 1.63TB
tier tier_enterprise
tier_capacity 0.00MB
tier_free_capacity 0.00MB
tier tier_nearline
tier_capacity 0.00MB
tier_free_capacity 0.00MB
has_nas_key no
layer replication
rc_buffer_size 48
compression_active no
compression_virtual_capacity 0.00MB
compression_compressed_capacity 0.00MB
compression_uncompressed_capacity 0.00MB
data_reduction yes
total_reclaimable_capacity 25.00MB
used_capacity_before_reduction
used_capacity_after_reduction
overhead_capacity
cache_prefetch on
email_organization
email_machine_address
email_machine_city
email_machine_state XX
email_machine_zip
email_machine_country
total_drive_raw_capacity 173.63TB
easy_tier_acceleration off
local_fc_port_mask:0000000000000000000000000000000000000000000000000000000000001101
partner_fc_port_mask:0000000000000000000000000000000000000000000000000000000000000011

topology:standard
topology_status:
compression_destage_mode false
vdisk_protection_time 120
vdisk_protection_enabled yes
product_name Compuhomer
odx on
max_replication_delay 333
partnership_exclusion_threshold 120

ibmcustomer 1234567
ibmcomponent SANVCNSW1
ibmcountry 866
tier_0_flash_compressed_data_used 0.00MB
tier_1_flash_compressed_data_used 0.00MB
```

```
tier_enterprise_compressed_data_used 0.00MB
tier_nearline_compressed_data_used 0.00MB
physical_capacity:0.00MB
physical_free_capacity:0.00MB
enhanced_callhome off
censor_callhome on
unmap off
```

A detailed invocation example

```
lssystem
```

The resulting output:

```
id 000002006C40A278
name cluster0
location local
partnership
bandwidth
total_mdisk_capacity 222.2GB
space_in_mdisk_grps 0
space_allocated_to_vdisks 0.00MB
total_free_space 222.2GB
total_vdiskcopy_capacity 0.00MB
total_used_capacity 0.00MB
total_overallocation 0
total_vdisk_capacity 0.00MB
total_allocated_extent_capacity 0.00MB
statistics_status on
statistics_frequency 15
cluster_locale en_US
time_zone 522 UTC
code_level 6.4.0.0 (build 61.9.1112130001)
console_IP 0.0.0.0:443
id_alias 000002006C40A278
gm_link_tolerance 300
gm_inter_cluster_delay_simulation 0
gm_intra_cluster_delay_simulation 0
gm_max_host_delay 5
email_reply
email_contact
email_contact_primary
email_contact_alternate
email_contact_location
email_contact2
email_contact2_primary
email_contact2_alternate
email_state stopped
inventory_mail_interval 0
cluster_ntp_IP_address
cluster_isns_IP_address
iscsi_auth_method none
iscsi_chap_secret
auth_service_configured no
auth_service_enabled no
auth_service_url
auth_service_user_name
auth_service_pwd_set no
auth_service_cert_set no
auth_service_type tip
relationship_bandwidth_limit 25
tier ssd
tier_capacity 0.00MB
tier_free_capacity 0.00MB
tier enterprise
tier_capacity 111.10GB
tier_free_capacity 111.10GB
tier nearline
```

```
tier_capacity 111.10GB
tier_free_capacity 111.10GB
has_nas_key no
layer replication
rc_buffer_size 48
compression_active no
compression_virtual_capacity 0.00MB
compression_compressed_capacity 0.00MB
compression_uncompressed_capacity 0.00MB
cache_prefetch on
email_organization
email_machine_address
email_machine_city
email_machine_state XX
email_machine_zip
email_machine_country
total_drive_raw_capacity 173.63TB
easy_tier_acceleration off
local_fc_port_mask:0000000000000000000000000000000000000000000000000000000000001101
partner_fc_port_mask:0000000000000000000000000000000000000000000000000000000000000011

topology:standard
topology_status:
compression_destage_mode false
vdisk_protection_time 120
vdisk_protection_enabled yes
product_name Compuhomer
odx on
max_replication_delay 333
deduplication_capacity_saving:100GB
overhead_capacity:23.00GB
```

lssystemcert

Use the **lssystemcert** command to list information about the current system Secure Sockets Layer (SSL) certificate.

Syntax

```
►►── lssystemcert ── ┬──────────┬─ ┬──────────────────────┬──────────────────────►◄
                     └─ -nohdr ─┘  └─ -delim ── delimiter ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default, in a concise view all columns of data are space-separated, with the width of each column set to the maximum width of each item of data. In a detailed view, each item of data is an individual row, and if you display headers, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. Enter -delim : on the command line, and the colon character (:) separates all items of data in a concise view (for example, the spacing of columns does not occur); in a detailed view, the specified *delimiter* separates the data from its header.

Description

This command lists information about the current system Secure Sockets Layer (SSL) certificate and indicates if there is an outstanding certificate request.

Table 49 provides the attribute values that can be displayed as output view data.

Table 49. lssystemcert output

Attribute	Possible Values
certificate	Indicates a readable version of the current SSL certificate.
certificate export	Indicates an encoded version of the SSL certificate.
certificate_request_outstanding	Indicates that there is an unfinished certificate request (if the value is yes) and installs a signed certificate. The value is yes or no.

An invocation example

```
lssystemcert
```

The detailed resulting output:

```
certificate: 58 fields
    Data:
        Version: 3 (0x2)
        Serial Number: 1431938814 (0x5559a6fe)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=US, L=Springfield, O=TMI, OU=ABC, CN=2154/emailAddress=chili@snpp.com
        Validity
            Not Before: May 18 08:46:54 2015 GMT
            Not After : May 14 08:46:54 2030 GMT
        Subject: C=US, L=Springfield, O=TMI, OU=ABC, CN=2154/emailAddress=chili@snpp.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:de:1c:70:c2:91:87:3c:6a:92:91:f7:d9:a3:5b:
                    05:e6:91:f1:87:c1:25:38:61:ad:4d:d9:26:19:7b:
                    9e:61:a5:fd:b1:d1:eb:d1:e4:a8:78:21:75:58:80:
                    4a:5c:dd:5e:6c:8b:1b:de:57:f9:d5:1f:71:92:3e:
                    78:d5:a4:75:1e:11:b2:62:18:52:0f:4d:32:a8:fd:
                    2b:16:4f:42:d1:d6:70:af:86:eb:fe:a1:ab:bc:66:
                    8a:44:bc:e0:36:53:77:96:2f:74:7d:95:33:79:c2:
                    59:5e:e1:43:50:da:43:25:c4:5d:3a:ac:d7:82:ad:
                    34:d5:ba:4c:52:4a:c0:81:3a:ad:e8:33:fe:4f:be:
                    e8:47:fa:5b:1f:dd:d8:9e:3b:44:a6:b6:b9:43:d2:
                    d4:45:8e:cb:5b:bb:10:5b:c9:30:68:2c:30:b6:e4:
                    ea:59:6d:a2:37:a7:13:77:28:1d:13:68:58:7b:dd:
                    90:d6:a8:81:7b:79:9f:1e:e4:a7:67:1b:7b:c5:b4:
                    90:dc:6b:d4:1f:7e:e9:e3:7b:ac:26:59:11:f1:99:
                    34:f0:6a:50:41:76:ad:a3:30:74:8f:8f:f5:ed:1e:
                    21:77:ff:51:90:1b:83:fb:04:f0:62:3d:71:17:a5:
                    ab:44:e8:bc:b0:82:0d:af:af:ae:68:5a:cf:e3:c8:
                    a9:53
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                87:66:33:16:61:7A:8E:CA:B4:BA:78:7B:56:56:8A:9D:C5:96:80:76
            X509v3 Authority Key Identifier:
                keyid:87:66:33:16:61:7A:8E:CA:B4:BA:78:7B:56:56:8A:9D:C5:96:80:76
    Signature Algorithm: sha256WithRSAEncryption
         56:b1:5d:59:11:ae:7b:6e:29:cc:1f:a8:75:77:d2:65:d6:88:
         75:8e:b9:cd:d6:71:ac:7e:89:8c:65:68:36:a8:28:97:88:36:
         42:da:a4:58:9b:c6:ce:c1:56:c9:0e:c5:ce:e7:01:74:d0:66:
         d0:4d:d3:0f:84:53:f6:e5:89:8e:44:6d:70:13:45:9c:21:91:
         50:f4:b0:b7:cc:cb:18:e8:d7:b3:38:b4:f5:5d:36:51:8c:7e:
         52:d4:24:0f:1f:2e:0a:b4:b6:9b:cb:23:43:6c:16:a2:a5:de:
         84:8a:0d:28:3c:d9:3d:5d:a4:52:44:28:90:98:a6:26:a9:c9:
         87:6c:27:3f:ef:09:5f:9d:0b:40:8d:07:64:ee:33:d9:40:47:
         98:02:10:58:2b:54:33:d9:37:69:d4:13:e6:0d:ec:46:26:b1:
         c1:c5:15:7c:8d:89:26:f7:95:d9:2f:d9:33:8c:f0:1a:dc:08:
         19:eb:18:16:51:30:a3:c0:ee:be:86:7d:3d:91:61:d5:99:bf:
         5e:19:b9:89:72:e1:4c:ea:5e:2b:90:ce:ce:75:83:e0:c9:14:
         83:21:21:e0:f8:28:94:90:71:e6:13:ca:97:8c:e3:58:b9:0c:
         62:03:e5:1c:1b:6c:dd:c3:60:48:d4:78:24:8e:22:34:78:32:
         fe:45:ee:36
certificate_export: 23 fields
-----BEGIN CERTIFICATE-----
MIIDzTCCArWgAwIBAgIEVVmm/jANBgkqhkiG9w0BAQsFADBqMQswCQYDVQQGEwJH
QjEQMA4GA1UEBwwHSHVyc2xleTEMMAoGA1UECgwDSUJNMQwwCgYDVQQLDANTU0cx
DTALBgNVBAMMBDIxNDUxHjAcBgkqhkiG9w0BCQEWD3N1cHBvcnRAaWJtLmNvbTAe
Fw0xNTA1MTgwODQ2NTRaFw0zMDA1MTQwODQ2NTRaMGoxCzAJBgNVBAYTAkdCMRAw
DgYDVQQHDAdIdXJzbGV5MQwwCgYDVQQKDANJQk0xDDAKBgNVBAsMA1NTRzENMAsG
A1UEAwwEMjE0NTEeMBwGCSqGSIb3DQEJARYPc3VwcG9ydEBpYm0uY29tMIIBIjAN
BgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA3hxwwpGHPGqSkffZo1sF5pHxh8El
OGGtTdkmGXueYaX9sdHr0eSoeCF1WIBKXN1ebIsb3lf51R9xkj541aR1HhGyYhhS
D00yqP0rFk9C0dZwr4br/qGrvGaKRLzgNlN3li90fZUzecJZXuFDUNpDJcRdOqzX
gq001bpMUkrAgTqt6DP+T77oR/pbH93YnjtEpra5Q9LURY7LW7sQW8kwaCwwtuTq
WW2iN6cTdygdE2hYe92Q1qiBe3mfHuSnZxt7xbSQ3GvUH37p43usJlkR8Zk08GpQ
QXatozB0j4/17R4hd/9RkBuD+wTwYj1xF6WrROi8sIINr6+uaFrP48ipUwIDAQAB
o3sweTAJBgNVHRMEAjAAMCwGCWCGSAGG+EIBDQQfFh1PcGVuU1NMIEdlbmVyYXRl
ZCBDZXJ0aWZpY2F0ZTAdBgNVHQ4EFgQUh2YzFmF6jsq0unh7VlaKncWWgHYwHwYD
VR0jBBgwFoAUh2YzFmF6jsq0unh7VlaKncWWgHYwDQYJKoZIhvcNAQELBQADggEB
AFaxXVkRrntuKcwfqHV30mXWiHWOuc3Wcax+iYxlaDaoKJeINkLapFibxs7BVskO
xc7nAXTQZtBN0w+EU/bliY5EbXATRZwhkVD0sLfMyxjo17M4tPVdNlGMflLUJA8f
Lgq0tpvLI0NsFqKl3oSKDSg82T1dpFJEKJCYpiapyYdsJz/vCV+dC0CNB2TuM9lA
R5gCEFgrVDPZN2nUE+YN7EYmscHFFXyNiSb3ldkv2TOM8BrcCBnrGBZRMKPA7r6G
fT2RYdWZv14ZuYly4UzqXiuQzs51g+DJFIMhIeD4KJSQceYTypeM41i5DGID5Rwb
bN3DYEjUeCSOIjR4Mv5F7jY=
-----END CERTIFICATE-----
certificate_request_outstanding:no
```

lssystemip

Use the **lssystemip** command to display a list of the clustered system (system) management IP addresses configured for each port.

Syntax

```
►►── lssystemip ── ┬─────────┬─ ── ┬────────────────────┬─ ───────────────────────►
                   └─ -nohdr ┘     └─ -delim ─ delimiter ┘

►─┬─────────────────────────────────┬─ ┬─────────────────┬─ ┬──────────────────────┬─ ────►
  └─ -filtervalue ─ attribute=value ┘  └─ -filtervalue? ┘  └─ -port ─ system_port ┘

►─┬─system_id───┬──────────────────────────────────────────────────────────►◄
  └─system_name─┘
```

Parameters

-nohdr

(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*

(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each data item has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (**:**) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-filtervalue *attribute=value*

(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed. If a capacity is specified, the units must also be included.

Note: Some filters allow the asterisk character (*) when you enter the command. The following rules apply to the use of wildcard characters when you use the command-line interface (CLI):

- The wildcard character is an asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, you must enclose the filter entry within double quotation marks (""), as shown in the following example:

```
lssystemip -filtervalue  "system_name=md*"
```

-filtervalue?

(Optional) displays a list of filters that can be applied against this view. The following filter attributes are valid for the **lssystemip** command:

- `port_id`
- `system_name`
- `system_id`

system_id | *system_name*

(Required) Specifies the name or ID of a system.

-port *system_port*

(Required) Specifies the system port (1 or 2) to apply changes to.

Description

This command displays a list of the system management IP addresses configured for each port.

Table 50 provides the attribute values that can be displayed as output view data.

Table 50. lssystemip output

Attribute	Possible Values
cluster_id	Indicates the ID of the system.
cluster_name	Indicates the name of the system.
location	Indicates the system location.
port_id	Indicates the ID of the port.

Table 50. lssystemip output (continued)

Attribute	**Possible Values**
IP_address	Indicates the Internet Protocol Version 4 (IPv4) address.
subnet_mask	Indicates the IPv4 subnet mask.
gateway	Indicates the IPv4 gateway.
IP_address_6	Indicates the Internet Protocol Version 6 (IPv6) address.
gateway_6	Indicates the IPv6 gateway.
prefix_6	Indicates the IPv6 prefix.

A concise invocation example

```
lssystemip -delim ,
```

The concise resulting output:

```
cluster_id,cluster_name,location,port_id,IP_address,subnet_mask,
gateway,IP_address_6,gateway_6,prefix_6
000002006CC0B71A,cl1,local,1,192.168.1.2,DHCP,255.255.255.0,192.168.1.1,
2001:0db8:85a3:0000:0000:8a2e:0370:7334,2001:0db8:85a3:0000:0000:8a2e:0370:7334,
2001:0db8:85a3:0000:0000:8a2e:0370:7334,64
000002006CC0B71A,cl1,local,2,192.168.1.2,DHCP,255.255.255.0,192.168.1.1,
2001:0db8:85a3:0000:0000:8a2e:0370:7334,2001:0db8:85a3:0000:0000:8a2e:0370:7334,
2001:0db8:85a3:0000:0000:8a2e:0370:7334,64
000002006CC0B7110,cl2,remote,1,192.168.1.2,DHCP,255.255.255.0,192.168.1.1,
2001:0db8:85a3:0000:0000:8a2e:0370:7334,2001:0db8:85a3:0000:0000:8a2e:0370:7334,
2001:0db8:85a3:0000:0000:8a2e:0370:7334,64
000002006CC0B7110,cl2,remote,2,192.168.1.2,DHCP,255.255.255.0,192.168.1.1,
2001:0db8:85a3:0000:0000:8a2e:0370:7334,2001:0db8:85a3:0000:0000:8a2e:0370:7334,
2001:0db8:85a3:0000:0000:8a2e:0370:7334,64
```

A detailed invocation example

```
lssystemip 000002006CC0B71A
```

The detailed resulting output:

```
cluster_id 000002006CC0B71A
cluster_name cl1
location local
port_id 1
IP_address 192.168.1.2
subnet_mask 255.255.255.0
gateway 192.168.1.1
IP_address_6 2001:0db8:85a3:0000:0000:8a2e:0370:7334
gateway_6 2001:0db8:85a3:0000:0000:8a2e:0370:7334
prefix_6 64

cluster_id 000002006CC0B71A
cluster_name cl1
location local
port_id 2
IP_address 192.168.1.2
subnet_mask 255.255.255.0
gateway 192.168.1.1
IP_address_6 2001:0db8:85a3:0000:0000:8a2e:0370:7334
gateway_6 2001:0db8:85a3:0000:0000:8a2e:0370:7334
prefix_6 64
```

lssystemstats

Use the **lssystemstats** command to display the most recent values of all node statistics in a clustered system (system), or to display a history of values for a specified subset of available statistics across all nodes in a system. This command also can be used to display a history of values for a specified subset of available statistics.

Syntax

```
►►— lssystemstats — ┬─────────┬─ ┬──────────────────────┬─ ──────────────────────────►
                    └─ -nohdr ┘  └─ -delim — delimiter ┘

►─┬───────────────────────────────────┬─ ┬─────────────────┬─ ┬─────────────────────┬─ ─────►◄
  └─ -filtervalue — attribute=value ┘  └─ -filtervalue? ┘  └─ -history stat_list ┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (**:**) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards:

- The wildcard character is the asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, enclose the filter entry within double quotation marks (""):

```
lssystemstats -filtervalue stat_name="io*"
```

-filtervalue?
(Optional) Displays the valid filter attributes for the **-filtervalue** *attribute=value* parameter:

- stat_name

-history stat_list
Provides the most recent node statistical values, specific node statistical values, or historical data for any node.

Description

This command returns one set of statistics for all the nodes in the system. The statistical values are determined by using samples that are received from each node.

Note: Values are rounded to the nearest integer when appropriate (for example, between one and ninety-nine for percentages).

Table 51 provides the attribute values that can be displayed as output view data.

Table 51. **`lssystemstats`** *attribute values*

Attribute	Value
stat_current	The current value of the statistic field.
stat_list	The system history of the reported statistics.
stat_name	The name of the statistic field.
stat_peak	The peak value of the statistic field in the last 5 minutes.
stat_peak_time	The time that the peak occurred.
sample_time	The time of the sample occurrence.
stat_value	The statistical value at the epoch interval.

Remember: Filtering is supported on the `stat_name` field by using the concise view.

A system summary invocation example

```
lssystemstats
```

The resulting output:

```
stat_name      stat_current stat_peak stat_peak_time
cpu_pc         5            6         111123104304
fc_mb          321          327       111123104129
fc_io          2167         2368      111123103904
sas_mb         438          534       111123104104
sas_io         5784         7738      111123104314
iscsi_mb       0            0         111123104359
iscsi_io       0            0         111123104359
write_cache_pc 0            0         111123104359
total_cache_pc 0            0         111123104359
vdisk_mb       321          326       111123104129
vdisk_io       2070         2276      111123103904
vdisk_ms       34           52        111123103954
mdisk_mb       320          329       111123104029
mdisk_io       3135         3340      111123103904
mdisk_ms       15           24        111123104314
drive_mb       440          534       111123104104
drive_io       5765         6572      111123104104
drive_ms       14           21        111123104314
vdisk_r_mb     174          178       111123104324
vdisk_r_io     1064         1180      111123103904
vdisk_r_ms     31           53        111123103954
vdisk_w_mb     146          159       111123104129
vdisk_w_io     1006         1160      111123104129
vdisk_w_ms     38           54        111123104314
mdisk_r_mb     172          177       111123104259
mdisk_r_io     2054         2184      111123103904
mdisk_r_ms     11           18        111123103954
mdisk_w_mb     146          160       111123104129
mdisk_w_io     1081         1229      111123104129
mdisk_w_ms     25           38        111123104314
drive_r_mb     207          356       111123104329
drive_r_io     2940         3952      111123104104
drive_r_ms     11           18        111123104314
drive_w_mb     231          250       111123104129
drive_w_io     2825         3156      111123104129
drive_w_ms     16           24        111123104314
iplink_mb      0            1         130711190446
```

```
iplink_io      0            10        130711190446
iplink_comp_mb      0           250        151014133723

cloud_up_mb    0            0         161118051715
cloud_up_ms    0            0         161118051715
cloud_down_mb  0            0         161118051715
cloud_down_ms  0            0         161118051715
```

A filtered system summary invocation example

```
lssystemstats -filtervalue stat_name=cpu_pc:stat_name=fc_mb -delim :
```

The resulting output:

```
The filtered system summary output:
stat_name:stat_current:stat_peak:stat_peak_time
cpu_pc:5:7:111123104547
fc_mb:319:339:111123104517
```

A historical view-based system summary invocation example

```
lssystemstats -history fc_io
```

The resulting partial output for the historical system summary example:

```
sample_time  stat_name stat_value
111123104224 fc_io     2120
111123104229 fc_io     2102
111123104234 fc_io     2041
111123104239 fc_io     2211
111123104244 fc_io     2204
111123104249 fc_io     2046
111123104254 fc_io     1997
111123104259 fc_io     2081
111123104304 fc_io     2123
111123104309 fc_io     2030
111123104314 fc_io     1754
111123104319 fc_io     1640
111123104324 fc_io     1759
111123104329 fc_io     1638
111123104334 fc_io     1804
111123104339 fc_io     2011
111123104344 fc_io     2028
111123104349 fc_io     2171
111123104354 fc_io     2055
111123104359 fc_io     2167
111123104404 fc_io     2140
111123104409 fc_io     2111
```

Table 52 provides the possible values that are applicable to the values that are displayed for the **stat_name** attribute.

Table 52. Stat_name field values

Value	Description
compression_cpu_pc	Displays the percentage of allocated CPU capacity that is used for compression.
cpu_pc	Displays the percentage of allocated CPU capacity that is used for the system.
fc_mb	Displays the total number of megabytes transferred per second (MBps) for Fibre Channel traffic on the system. This value includes host I/O and any bandwidth that is used for communication within the system.
fc_io	Displays the total input/output (I/O) operations that are transferred per seconds for Fibre Channel traffic on the system. This value includes host I/O and any bandwidth that is used for communication within the system.

Table 52. Stat_name field values (continued)

Value	Description
sas_mb	Displays the total number of megabytes transferred per second (MBps) for serial-attached SCSI (SAS) traffic on the system. This value includes host I/O and bandwidth that is used for background RAID activity.
sas_io	Displays the total I/O operations that are transferred per second for SAS traffic on the system. This value includes host I/O and bandwidth that is used for background RAID activity.
iscsi_mb	Displays the total number of megabytes transferred per second (MBps) for iSCSI traffic on the system.
iscsi_io	Displays the total I/O operations that are transferred per second for iSCSI traffic on the system.
write_cache_pc	Displays the percentage of the write cache usage for the node.
total_cache_pc	Displays the total percentage for both the write and read cache usage for the node.
vdisk_mb	Displays the average number of megabytes transferred per second (MBps) for read and write operations to volumes during the sample period. **Note:** Only the write operation value is displayed.
vdisk_io	Displays the average number of I/O operations that are transferred per second for read and write operations to volumes during the sample period.
vdisk_ms	Displays the average amount of time in milliseconds that the system takes to respond to read and write requests to volumes over the sample period.
mdisk_mb	Displays the average number of megabytes transferred per second (MBps) for read and write operations to MDisks during the sample period.
mdisk_io	Displays the average number of I/O operations that are transferred per second for read and write operations to MDisks during the sample period.
mdisk_ms	Displays the average amount of time in milliseconds that the system takes to respond to read and write requests to MDisks over the sample period.
drive_mb	Displays the average number of megabytes transferred per second (MBps) for read and write operations to drives during the sample period
drive_io	Displays the average number of I/O operations that are transferred per second for read and write operations to drives during the sample period.
drive_ms	Displays the average amount of time in milliseconds that the system takes to respond to read and write requests to drives over the sample period.
vdisk_w_mb	Displays the average number of megabytes transferred per second (MBps) for read and write operations to volumes during the sample period.
vdisk_w_io	Displays the average number of I/O operations that are transferred per second for write operations to volumes during the sample period.
vdisk_w_ms	Displays the average amount of time in milliseconds that the system takes to respond to write requests to volumes over the sample period.
mdisk_w_mb	Displays the average number of megabytes transferred per second (MBps) for write operations to MDisks during the sample period.
mdisk_w_io	Displays the average number of I/O operations that are transferred per second for write operations to MDisks during the sample period.
mdisk_w_ms	Displays the average amount of time in milliseconds that the system takes to respond to write requests to MDisks over the sample period.
drive_w_mb	Displays the average number of megabytes transferred per second (MBps) for write operations to drives during the sample period
drive_w_io	Displays the average number of I/O operations that are transferred per second for write operations to drives during the sample period.

Table 52. Stat_name field values (continued)

Value	Description
drive_w_ms	Displays the average amount of time in milliseconds that the system takes to respond write requests to drives over the sample period.
vdisk_r_mb	Displays the average number of megabytes transferred per second (MBps) for read operations to volumes during the sample period.
vdisk_r_io	Displays the average number of I/O operations that are transferred per second for read operations to volumes during the sample period.
vdisk_r_ms	Displays the average amount of time in milliseconds that the system takes to respond to read requests to volumes over the sample period.
mdisk_r_mb	Displays the average number of megabytes transferred per second (MBps) for read operations to MDisks during the sample period.
mdisk_r_io	Displays the average number of I/O operations that are transferred per second for read operations to MDisks during the sample period.
mdisk_r_ms	Displays the average amount of time in milliseconds that the system takes to respond to read requests to MDisks over the sample period.
drive_r_mb	Displays the average number of megabytes transferred per second (MBps) for read operations to drives during the sample period
drive_r_io	Displays the average number of I/O operations that are transferred per second for read operations to drives during the sample period.
drive_r_ms	Displays the average amount of time in milliseconds that the system takes to respond to read requests to drives over the sample period.
iplink_mb	Displays the average number of megabytes requested to be transferred per second (MBps) over the IP partnership link during the sample period. This value is calculated before any compression of the data takes place. This value does not include iSCSI host input/output (I/O) operations.
iplink_comp_mb	Displays the average number of compressed megabytes transferred per second (MBps) over the IP Replication link during the sample period. This value is calculated after any compression of |the data takes place. This value does not include iSCSI host I/O operations. **Note:** If compression is disabled, the `iplink_mb` `stats` ID value is displayed instead.
cloud_up_mb	Displays the average number of megabytes transferred per second (Mbps) for upload operations to a cloud account during the sample period.
cloud_up_ms	Displays the average amount of time (in milliseconds) it takes for the system to respond to upload requests to a cloud account during the sample period.
cloud_down_mb	Displays the average number of Mbps for download operations to a cloud account during the sample period.
cloud_down_ms	Displays the average amount of time (in milliseconds) it takes for the system to respond to download requests to a cloud account during the sample period.

lstargetportfc

Use the **`lstargetportfc`** command to generate the lists of worldwide port names (WWPNs) required to set up Fibre Channel (FC) zoning and to display the current failover status of host I/O ports.

Syntax

```
►►— lstargetportfc — ┬─────────────────────────────────┬─ ┬──────────┬─ ──────────────────►
                     └─ -filtervalue — attribute=value ─┘  └─ -nohdr ─┘
```

```
►─┬─────────────────────┬─ ─ ─┬──────────────────┬──────────────────────────────────────────►◄
  └─ -delim ── delimiter ─┘     └─ -filtervalue? ─┘
```

Parameters

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are returned. If a capacity is specified, the units must also be included.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcard characters when you use the CLI:

- The wildcard character is an asterisk (*), which must be the first or last character in the string.
- The command can contain a maximum of one wildcard.
- When you use a wildcard, enclose the filter entry within double quotation marks ("").

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If no data is displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-filtervalue?
(Optional) Displays the valid filter attributes for the **lstargetportfc** command:

- `port_id`
- `owning_node_id`
- `current_node_id`
- `host_io_permitted`
- `virtualized`

Description

This command generates lists of worldwide port names (WWPNs) required to set up Fibre Channel (FC) zoning. This command also displays the current failover status of host I/O ports.

Table 53 provides the attribute values that can be displayed as output view data.

Table 53. lstargetportfc output

Attribute	Description
id	Indicates the ID of the port.
WWPN	Indicates the WWPN of the port. The value is hexadecimal.
WWNN	Indicates the worldwide node name (WWNN) of the port. The value is hexadecimal.
port_id	Indicates the system port ID. The value is the same as the **lsportfc** `port_id` field.

Table 53. lstargetportfc output (continued)

Attribute	Description
owning_node_id	Indicates the ID of the node that owns the port. **Note:** This node can be offline whether the port is online or offline.
current_node_id	Indicates the ID of the node on which this port is active. The value is blank if the port is not active on any node
nportid	Indicates the `nportid` hexadecimal value.
host_io_permitted	Indicates whether host I/O operations can run on the port. The values are `yes` and `no`.
virtualized	Indicates whether it is a virtualized port. The values are `yes` and `no` (which indicates that this port cannot be online on any node other than the owning node).

An invocation example

This example shows a single I/O group with two nodes. One 2-port FC card is installed on each node, and the I/O group's `fctargetportmode` setting is set to `disabled`.

```
lstargetportfc
```

The detailed resulting output:

```
id WWPN             WWNN             port_id owning_node_id current_node_id nportid host_io_permitted     virtualized
1  500507680140BADD 500507680100BADD 1       1              1               0E2411  yes                   no
2  500507680141BADD 500507680100BADD 1       1                              000000  no                    yes
3  500507680130BADD 500507680100BADD 2       1              1               0E2412  yes                   no
4  500507680131BADD 500507680100BADD 2       1                              000000  no                    yes
5  500507680140BADE 500507680100BADE 1       2              2               0E2413  yes                   no
6  500507680141BADE 500507680100BADE 1       2                              000000  no                    yes
7  500507680130BADE 500507680100BADE 2       2              2               0E2414  yes                   no
8  500507680131BADE 500507680100BADE 2       2                              000000  no                    yes
```

An invocation example

This example shows a single I/O group with two nodes. One 2-port FC card is installed on each node, and the I/O group's `fctargetportmode` setting is set to `transitional`.

```
lstargetportfc
```

The detailed resulting output:

```
id WWPN             WWNN             port_id owning_node_id current_node_id nportid host_io_permitted     virtualized
1  500507680140BADD 500507680100BADD 1       1              1               0E2411  yes                   no
2  500507680141BADD 500507680100BADD 1       1              1               0E2412  yes                   yes
3  500507680130BADD 500507680100BADD 2       1              1               0E2413  yes                   no
4  500507680131BADD 500507680100BADD 2       1              1               0E2414  yes                   yes
5  500507680140BADE 500507680100BADE 1       2              2               0E2415  yes                   no
6  500507680141BADE 500507680100BADE 1       2              2               0E2416  yes                   yes
7  500507680130BADE 500507680100BADE 2       2              2               0E2417  yes                   no
8  500507680131BADE 500507680100BADE 2       2              2               0E2418  yes                   yes
```

(satask) mkcluster

Use the **`mkcluster`** command to create a new clustered system (system).

Syntax

```
►►— satask —— mkcluster —— -clusterip — -ipv4_ip — ————————————————————————►
```

```
►─ -gw ── -ipv4_gw ── -mask ── ──ipv4_mask ── ┬──────────────────────┬─ ┬──────────────┬─ ───────►◄
                                              └─ -name ──cluster_name┘  └─ panel_name ─┘

►►── satask ── ── mkcluster ── ── -clusterip_6 ── ── ipv6_ip ── ──────────────────────────────────►

►─ -gw_6 ── ipv6_gw ── -prefix_6 ── ── ipv6_subnet ── ───────────────────────────────────────────►

►┬──────────────────────┬─ ┬──────────────┬─ ─────────────────────────────────────────────────────►◄
 └─ -name ──cluster_name┘  └─ panel_name ─┘
```

Parameters

-clusterip *ipv4_ip*
(Optional) The Internet Protocol Version 4 (IPv4) address for system Ethernet port 1.

-gw *ipv4_gw*
(Optional) The IPv4 gateway for system Ethernet port 1.

-mask *ipv4_mask*
(Optional) The IPv4 subnet for system Ethernet port 1.

-clusterip_6 *ipv6_ip*
(Optional) The Internet Protocol Version 6 (IPv6) address for system Ethernet port 1.

-gw_6 *ipv6_gw*
(Optional) The IPv6 gateway for system Ethernet port 1.

-prefix_6 *ipv6_subnet*
(Optional) The IPv6 prefix for system Ethernet port 1.

-name *cluster_name*
(Optional) The name of the new system.

panel_name
(Optional) Identifies the node being serviced.

Note: If panel_name is not supplied, this applies to the node on which the command is running.

Remember: You must specify one of the following:

- IPv4 system IP, gateway and subnet
- IPv6 system IP, gateway, and prefix

Description

This command creates a new system.

An invocation example using specific -clusterip, -gw, and -mask parameters

```
satask mkcluster -clusterip 192.168.1.2 -gw 192.168.1.1 -mask 255.255.255.0
```

The resulting output:

```
No feedback
```

mkcluster (Deprecated)

The **mkcluster** system command is deprecated. Use the **satask mkcluster** command to create a new clustered system (system).

mkquorumapp

Use the **mkquorumapp** command to generate a Java™ application to use for quorum.

Syntax

```
►►── mkquorumapp ── ┬─────────┬ ─────────────────────────────────────────────►◄
                    └─ -ip_6 ─┘
```

Parameters

-ip_6
(Optional) Specifies that the quorum application use IPv6 service addresses to connect to nodes.

Description

This command generates a Java application to use for quorum.

An invocation example

This example creates the file `/dumps/ip_quorum.jar` to use for IP quorum on an IPv4 network.

```
mkquorumapp
```

The detailed resulting output:

```
No feedback
```

An invocation example

This example creates the file `/dumps/ip_quorum.jar` to use for IP quorum on an IPv6 network.

```
mkquorumapp -ip_6:
```

The detailed resulting output:

mkthrottle

Use the **mkthrottle** command to create a new throttle object and associate it with an object (such as a volume). You can also create offloaded I/O throttling (which is a single clustered system throttle).

Syntax

```
►►── mkthrottle ── ── -type ── offload | vdisk | host | hostcluster | mdiskgrp ── ──────────────►

►─┬──────────────────────────────────────┬ ┬────────────────────────┬ ──────────────────────────►
  └─ -bandwidth ── bandwidth_limit_in_mb ┘ └─ -iops ── iops_limit ──┘

►─┬─────────────────────────┬ ┬───────────────────────────┬ ┬──────────────────────────┬────────►
  └─ -name ── throttle_name ┘ └─ -vdisk ──┬─ vdisk_id ───┬┘ └─ -host ──┬─ host_id ───┬┘
                                          └─ vdisk_name ─┘            └─ host_name ─┘

►─ ┬─────────────────────────────────────────┬ ┬────────────────────────────────────┬──────────►◄
   └─ -hostcluster ──┬─ hostcluster_id ───┬──┘ └─ -mdiskgrp ──┬─ mdiskgrp_id ───┬──┘
                     └─ hostcluster_name ─┘                   └─ mdiskgrp_name ─┘
```

Parameters

-type *offload* | *vdisk* | *host* | *hostcluster* | *mdiskgrp*
(Required) Specifies the type of throttle, either *offload* or *vdisk*.

-bandwidth *bandwidth_limit_in_mb*
(Optional) Specifies the bandwidth in MBps. This must be a numeric value 0 - 268435456.

-iops *iops_limit*
(Optional) Specifies the I/O operations limit. This must be a numeric value 0 - 33554432.

-name *throttle_name*
(Optional) Specifies the throttling object's name. This value must be an alphanumeric string up to 63 characters long.

-vdisk *vdisk_id* | *vdisk_name*
(Optional) Specifies the volume ID or name of the volume to throttle. The value must be a numeric or alphanumeric string.

Note: This keyword must be specified when you specify -type *vdisk*.
This parameter is mandatory for volume throttling but cannot be used for offload throttling.

-host *host_id* | *host_name*
(Optional) Specifies the host ID or name to throttle.

-hostcluster *hostcluster_id* | *hostcluster_name*
(Optional) Specifies the host cluster ID or name to throttle.

-mdiskgrp *mdiskgrp_id* | *mdiskgrp_name*
(Optional) Specifies the MDisk group (storage pool) or name to throttle. This applies to parent or child storage pools.

Description

This command creates a new throttle object and associates it with an object (such as a volume).

Note:

- A throttle object cannot be defined for a host if it is a part of host cluster that already has a host cluster throttle object defined for it.
- If a host cluster does not have a throttle object defined, its member hosts can have individual host throttles defined.
- The storage pool throttle objects for a child pool and a parent pool work independently of each other
- If a volume has multiple copies then throttling is done for the storage pool serving primary copy. Throttling is not applicable for secondary pools that are part of mirrored volumes or stretched cluster implementations.
-

An invocation example for creating a volume throttle of 10000 IOPs and a bandwidth limit of 500 MBps for volume vdisk0

```
mkthrottle -type vdisk -iops 10000 -bandwidth 500 -vdisk vdisk0
```

The detailed resulting output:

```
No feedback
```

An invocation example for creating offloaded I/O throttling with a bandwidth limit of 500 MBps

```
mkthrottle -type offload -bandwidth 500
```

The detailed resulting output:

```
No feedback
```

An invocation example for creating a host with a bandwidth limit of 100 MBps

```
mkthrottle -type host -bandwidth 100 -host host_Win2012SP2
```

The detailed resulting output:

```
No feedback
```

An invocation example for creating a host cluster with a bandwidth limit of 3000 MBps

```
mkthrottle -type hostcluster -bandwidth 3000 -hostcluster 0
```

The detailed resulting output:

```
No feedback
```

An invocation example for creating an MDisk group with a bandwidth limit of 4000 MBps

```
mkthrottle -type mdiskgrp -bandwidth 4000 -mdiskgrp 0
```

The detailed resulting output:

```
No feedback
```

ping

Use the **ping** command to diagnose IP configuration problems by checking whether the specified IP address is accessible from the node on which the command is run using the specified IP address.

Syntax

```
►►— ping ——┬— -srcip4 —source_ipv4_address destination_ipv4_address—┬——————————————————————————————►◄
           └— -srcip6 —source_ipv6_address destination_ipv6_address—┘
```

Parameters

-srcip4 *source_ipv4_address destination_ipv4_address*
(Required if **-srcip6** is not specified) Specifies the IPv4 address that sends the ping packet. The IPv4 address must already be bound to a port on the node on which the command is issued. If you do not specify this parameter you must specify **srcip6**.

-srcip6 *source_ipv6_address destination_ipv6_address*
(Required if **-srcip4** is not specified) Specifies the IPv6 address that sends the ping packet. The IPv6 address must already be bound to a port on the node on which the command is issued. If you do not specify this parameter you must specify **srcip4**.

Description

This command checks whether the specified IP address is accessible from the node on which the command is run using the specified IP address.

Use this command to ping from any port on any node as long as you are logged on to the service assistant on that node.

An invocation example

```
ping -srcip4 192.168.1.51 192.168.1.30
```

The resulting output:

```
PING 192.168.1.51 (192.168.1.51)PING 9.20.136.11 (9.20.136.11) 56(84) bytes of data.
64 bytes from 192.168.1.51: icmp_seq=1 ttl=249 time=0.690 ms
64 bytes from 192.168.1.51: icmp_seq=2 ttl=249 time=0.382 ms
64 bytes from 192.168.1.51: icmp_seq=3 ttl=249 time=0.311 ms

PING 192.168.1.30 (192.168.1.30)PING 9.20.136.11 (9.20.136.11) 56(84) bytes of data.
64 bytes from 192.168.1.30: icmp_seq=1 ttl=249 time=0.690 ms
64 bytes from 192.168.1.30: icmp_seq=2 ttl=249 time=0.382 ms
64 bytes from 192.168.1.30: icmp_seq=3 ttl=249 time=0.311 ms
```

rmiscsistorageport

Use the **rmiscsistorageport** command to remove established Internet Small Computer Systems Interface (iSCSI) sessions between system nodes and backend iSCSI target.

Syntax

```
►►— rmiscsistorageport — — lsiscsistorageport_row_id — ——————————————————————————————►◄
```

Parameters

lsiscsistorageport_row_id
(Required) Specifies the row ID of the selected row in the output of **lsiscsistorageport** command.

Description

Use this command to remove path groups that are established after you specify **addiscsistorageport** (not individual paths).

Any dependencies must be removed before you specify this command. The ID of the session that is listed (after you specify **lsiscsistorageport**) is used to identify the sessions that must be removed.

A detailed invocation example

First, specify **addiscsistorageport** for discovery and **lsiscsistorageport** to show any added session. Then, specify **rmiscsistorageport** to remove sessions that are indicated by view ID 0. No *tgt_user_name* or *target_chap* is required for discovery or session establishment.

```
rmiscsistorageport 0
```

The following detailed output is displayed:

```
No feedback
```

rmnode (SVC) / rmnodecanister (Storwize family products)

The **rmnode** / **rmnodecanister** command deletes a node from the clustered system. You can enter this command any time after a clustered system has been created.

Syntax

```
►►— rmnode — | rmnodecanister — ┬─────────┬— — -deactivatespare — ┬─ object_id ───┬──────────►◄
                                 └ -force ─┘                       └─ object_name ─┘
```

Parameters

-force
(Optional) Overrides the checks that this command runs. The parameter overrides the following two checks:

- If the command results in volumes going offline, the command fails unless the **force** parameter is used.
- If the command results in a loss of data because there is unwritten data in the write cache that is contained only within the node or node canister to be removed, the command fails unless the **force** parameter is used.

If you use the **force** parameter as a result of an error about volumes going offline, you force the node or node canister removal and run the risk of losing data from the write cache. The **force** parameter should always be used with caution.

-deactivatespare
(Optional) Specifies that the spare node (for this node) must be deactivated.

Important: Do not remove an offline node while a spare node is active.

object_id | *object_name*
(Required) Specifies the object name or ID that you want to modify. The variable that follows the parameter is either:

- The object name that you assigned when you added the node to the clustered system
- The object ID that is assigned to the node (not the worldwide node name)

Description

This command removes a node or node canister from the clustered system. This makes the node or node canister a candidate to be added back into this clustered system or into another system. After the node or node canister is deleted, the other node in the I/O group enters write-through mode until another node or node canister is added back into the I/O group.

Attention: When you run the **rmnode** command to remove the configured hardware for a node:

- Small Computer System Interface-3 (SCSI-3) reservations (through that node) are removed
- Small Computer System Interface-3 (SCSI-3) registrations (through that node) are removed

By default, the **rmnode** / **rmnodecanister** command flushes the cache on the specified node before the node or node canister is taken offline. In some circumstances, such as when the system is already degraded (for example, when both nodes in the I/O group are online and the virtual disks within the I/O group are degraded), the system ensures that data loss does not occur as a result of deleting the only node or node canister with the cache data.

The cache is flushed before the node or node canister is deleted to prevent data loss if a failure occurs on the other node or node canister in the I/O group.

To take the specified node or node canister offline immediately without flushing the cache or ensuring data loss does not occur, run the **rmnode** / **rmnodecanister** command with the **-force** parameter.

Prerequisites:

Before you issue the **rmnode** / **rmnodecanister** command, perform the following tasks and read the following **Attention** notices to avoid losing access to data:

1. Determine which virtual disks (VDisks, or volumes) are still assigned to this I/O group by issuing the following command. The command requests a filtered view of the volumes, where the filter attribute is the I/O group.

   ```
   lsvdisk -filtervalue IO_group_name=name
   ```

   where *name* is the name of the I/O group.
2. Determine the hosts that the volumes are mapped to by issuing the **lsvdiskhostmap** command.
3. Determine if any of the volumes that are assigned to this I/O group contain data that you need to access:
   - If you do not want to maintain access to these volumes, go to step 5.
   - If you do want to maintain access to some or all of the volumes, back up the data or migrate the data to a different (online) I/O group.
4. Determine if you need to turn the power off to the node or node canister:
   - If this is the last node or node canister in the clustered system, you do not need to turn the power off to the node or node canister. Go to step 5.
   - If this is not the last node or node canister in the cluster, turn the power off to the node or node canister that you intend to remove. This step ensures that the Subsystem Device Driver (SDD) does not rediscover the paths that are manually removed before you issue the delete node or node canister request.
5. Update the SDD configuration for each virtual path (vpath) that is presented by the volumes that you intend to remove. Updating the SDD configuration removes the vpaths from the volumes. Failure to update the configuration can result in data corruption. See the *Multipath Subsystem Device Driver: User's Guide* for details about how to dynamically reconfigure SDD for the given host operating system.
6. Quiesce all I/O operations that are destined for the node or node canister that you are deleting. Failure to quiesce the operations can result in failed I/O operations being reported to your host operating systems.

Attention:

1. Removing the last node in the cluster destroys the clustered system. Before you delete the last node or node canister in the clustered system, ensure that you want to destroy the clustered system.
2. If you are removing a single node or node canister and the remaining node or node canister in the I/O group is online, the data can be exposed to a single point of failure if the remaining node or node canister fails.
3. This command might take some time to complete since the cache in the I/O group for that node or node canister is flushed before the node or node canister is removed. If the **-force** parameter is used, the cache is not flushed and the command completes more quickly. However, if the deleted node or node canister is the last node or node canister in the I/O group, using the **-force** option results in the write cache for that node or node canister being discarded rather than flushed, and data loss can occur. The **-force** option should be used with caution.
4. If both nodes or node canisters in the I/O group are online and the volumes are already degraded before deleting the node or node canister, redundancy to the volumes is already degraded and loss of access to data and loss of data might occur if the **-force** option is used.

Notes:

1. If you are removing the configuration node or node canister, the **rmnode** / **rmnodecanister** command causes the configuration node or node canister to move to a different node or node canister within the clustered system. This process might take a short time: typically less than a minute. The clustered

system IP address remains unchanged, but any SSH client attached to the configuration node or node canister might need to reestablish a connection. The management GUI reattaches to the new configuration node or node canister transparently.

2. If this is the last node or node canister in the clustered system or if it is currently assigned as the configuration node, all connections to the system are lost. The user interface and any open CLI sessions are lost if the last node or node canister in the clustered system is deleted. A time-out might occur if a command cannot be completed before the node or node canister is deleted.

An invocation example for rmnode

```
rmnode 1
```

The resulting output:

```
No feedback
```

An invocation example for rmnodecanister

```
rmnodecanister 1
```

The resulting output:

```
No feedback
```

An invocation example

```
rmnode -deactivatespare
```

The resulting output

```
No feedback
```

rmportip

Use the **rmportip** command to remove an internet Small Computer System Interface (iSCSI) Internet Protocol (IP) address from a node Ethernet port.

Syntax

```
►►── rmportip ── ┬─────────────┬── ┬────────┬── -node ──┬─ node_name ─┬── ──port_id── ──────────►◄
                 └─ -failover ─┘   └─ -ip_6 ─┘          └─ node_id ───┘
```

Parameters

-failover
(Optional) Specifies that the failover IP address information is removed for the specified port.

-ip_6
(Optional) Specifies that the Internet Protocol Version 6 (IPv6) address is removed for the specified port. If this parameter is not used, the Internet Protocol Version 4 (IPv4) address is removed by default.

-node *node_name* | *node_id*
(Required) Specifies the node with the Ethernet port that the IP address is being removed from.

port_id
(Required) Specifies which port (1, 2, 3, or 4) to apply changes to.

Description

This command removes an IPv4 or IPv6 address from an Ethernet port of a node.

Before you unconfigure an IP from a source Ethernet port, the system checks whether any sessions are established from the selected port to any back-end iSCSI controller. You must remove sessions to the back-end iSCSI controller by using the command before you attempt to unconfigure the port. When the last IP (IPv4 or IPv6) address on a port is removed, the host port group ID that is associated with an iSCSI port is removed.

An invocation example for IPv4

```
rmportip -node 1   1
```

The resulting output:

```
No feedback
```

An invocation example for IPv6

```
rmportip -node 1 -ip_6 2
```

The resulting output:

```
No feedback
```

rmthrottle

Use the **rmthrottle** command to remove the throttle object associated with any volume.

Syntax

```
►►— rmthrottle — ─┬─ throttle_id ───┬──────────────────────────────────────►◄
                  └─ throttle_name ─┘
```

Parameters

throttle_id | *throtle_name*
(Required) Specifies the throttle object ID or name. The value must be a numeric or alphanumeric string up to 63 characters long.

Description

This command removes the throttle object associated with a specified volume.

An invocation example to remove a throttle object with the ID 2

```
rmthrottle 2
```

The detailed resulting output:

```
No feedback
```

An invocation example to remove a throttle object with name throttle_vdisk2

```
rmthrottle throttle_vdisk2
```

The detailed resulting output:

```
No feedback
```

setclustertime (Discontinued)

Attention: The **setclustertime** command has been discontinued. Use the **setsystemtime** command instead.

setsystemtime

Use the **setsystemtime** command to set the time for the clustered system (system).

Syntax

```
►►— setsystemtime — — -time — time_value ——————————————————————————————————————►◄
```

Parameters

-time *time_value*
: (Required) Specifies the time to which the system must be set. This must be in the following format (where M is month, D is day, H is hour, m is minute, and Y is year):

```
MMDDHHmmYYYY
```

Description

This command sets the time for the a system.

An invocation example

```
setsystemtime -time 040509142003
```

The resulting output:

```
No feedback
```

setpwdreset

Use the **setpwdreset** command to view and change the status of the password-reset feature for the node.

Syntax

```
►►— setpwdreset — ——┬— -disable —┬——————————————————————————————————————————————►◄
                    ├— -enable ——┤
                    └— -show ————┘
```

Parameters

-disable
: Disables the password-reset feature that is available through the front panel menu system.

-enable
: Enables the password-reset feature that is available through the front panel menu system.

-show
: Displays the status of the password-reset feature, which is either `enabled` or `disabled`.

Description

The system provides an option to reset the system superuser password to the default value. Use the front panel menu system.

This command allows access if the system superuser password is forgotten. If this feature remains enabled, make sure there is adequate physical security to the system hardware.

You can view or change the status of this feature.

An invocation example

```
setpwdreset -show
```

The resulting output:

```
Password status: [1]
```

This output means that the password or reset feature that is available through the front panel menu system is enabled. If the password status is `[0]`, this feature is disabled.

settimezone

Use the **settimezone** command to set the time zone for the clustered system (system).

Syntax

```
►►── settimezone ── ── -timezone ── timezone_arg ──────────────────────────────────────►◄
```

Parameters

-timezone *timezone_arg*
Specifies the time zone to set for the system.

Description

(Optional) This command sets the time zone for the system. Use the **-timezone** parameter to specify the numeric ID of the time zone that you want to set. Issue the **lstimezones** command to list the time-zones that are available on the system. A list of valid time-zones settings is displayed in a list.

Set the time zone to use when formatting the event log that is produced by issuing `dumperrlog`

Issue the **showtimezone** command to display the current time-zone settings for the system. The system ID and its associated time-zone are displayed. Issue the **setsystemtime** command to set the time for the system.

An invocation example

```
settimezone -timezone 5
```

The resulting output:

```
No feedback
```

showtimezone

Use the **showtimezone** command to display the current time zone settings for the cluster.

Syntax

```
►►── showtimezone ── ─┬──────────┬─ ─┬────────────────────┬──────────────────────────►◄
                      └─ -nohdr ─┘   └─ -delim ── delimiter ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum possible width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a one-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified character.

Description

This command displays a single time zone and its associated ID. This is the current time zone setting for the cluster. A list of available time-zones can be viewed by running the **lstimezones** command. The time zone can be changed by running the **settimezone** command.

An invocation example

```
showtimezone -delim :
```

The resulting output:

```
id:timezone
522:UTC
```

startstats

Use the **startstats** command to modify the interval at which per-node statistics for volumes, managed disks (MDisks), and nodes are collected.

Syntax

```
►►── startstats ── ── -interval ── time_in_minutes ──────────────────────────────────────►◄
```

Parameters

-interval *time_in_minutes*
(Required) Specifies the time in minutes. This time interval is the interval between the gathering of statistics, 1 - 60 minutes in increments of 1 minute.

Description

Running the **startstats** command resets the statistics timer to zero, and give it a new interval at which to sample. Statistics are collected at the end of each sampling period as specified by the **-interval** parameter. These statistics are written to a file, with a new file created at the end of each sampling period. Separate files are created for MDisks, volumes, and node statistics.

The files generated are written to the `/dumps/iostats` directory.

A maximum of 16 files are stored in the directory at any one time for each statistics file type, for example:

```
Nm_stats_nodepanelname_date_time
Nv_stats_nodepanelname_date_time
Nn_stats_nodepanelname_date_time
```

Statistics files are created for all time intervals. Before the 17th file for each type is created, the oldest file of that type is deleted.

These files can be listed by using the **lsdumps** command.

The following naming convention is used for these files:

*stats_type*_stats_*nodepanelname_date_time*

Where:

- The value for *stats_type* is `Nm` for MDisks, `Nv` for volumes, and `Nn` for node statistics.
- The value for *nodepanelname* is the current configuration node panel name.
- The value for *date* is in the format of `yymmdd`.
- The value for *time* is in the format of `hhmmss`.

This an example of:

- An MDisk statistics file name: `Nm_stats_000229_031123_072426`
- A volume statistics file name: `Nv_stats_000229_031123_072426`
- A node statistics file name: `Nn_stats_000229_031123_072426`

Statistics are collected for each MDisk and recorded in the `Nm_stats_`*`nodepanelname_date_time`* file, including the following statistical information:

- The number of SCSI read and write commands that are processed during the sample period
- The number of blocks of data that is read and written during the sample period
- Per MDisk, cumulative read and write external response times in milliseconds
- Per MDisk, cumulative read and write queued response times

Statistics are collected for each volume and recorded in the `Nv_stats_`*`nodepanelname_date_time`* file, including the following statistical information:

- The total number of processed SCSI read and write commands
- The total amount of read and written data
- Cumulative read and write response time in milliseconds
- Statistical information about the read or write cache usage
- Mirroring statistics that include latency

Statistics are collected for the node from which the statistics file originated, and those statistics are recorded in the `Nn_stats_`*`nodepanelname_date_time`* file, including the following statistical information:

- The usage figure for the node from which the statistic file was obtained
- The amount of data that is transferred to and received from each port on the node to other devices on the SAN
- Any statistical information about communication to other nodes on the fabric

An invocation example

```
startstats -interval 25
```

The resulting output:

```
No feedback
```

stopstats (Deprecated)

The **stopstats** command is deprecated. You can no longer disable statistics collection.

stopcluster (Discontinued)

Attention: The **stopcluster** command has been discontinued. Use the **stopsystem** command instead.

stopsystem

Use the **stopsystem** command to shut down a single node or the entire clustered system in a controlled manner. When you issue this command, you are prompted with a confirmation of intent to process the command.

Syntax

```
►►── stopsystem ── ── -reboot ── ─┬────────┬─ ─┬───────────────────────┬──────────────►◄
                                  └─ -force ┘  └─ -node ─┬─ node_name ─┤
                                                         └─ node_id ───┘
```

Parameters

-force
(Optional) Specifies that the node that is being shut down is the last online node in a given I/O group. The **-force** parameter also overrides the checks that this command runs. The parameter overrides the following two checks:

- If the command results in volumes going offline, the command fails unless the **-force** parameter is used.
- If the node being shut down is the last online node in the I/O group, the command fails unless the **-force** parameter is used.

If you use the **-force** parameter as a result of an error about volumes going offline, you force the node to shut down, even if it is the last online node in the I/O group.

Remember: The **-force** parameter should always be used with caution.

-node *node_name* | *node_id*
(Optional) Specifies the node that you want to shut down. You can specify one of the following values:

- The node name, or label that you assigned when you added the node to the system.
- The node ID that is assigned to the node (not the worldwide node name).

If you specify **-node** *node_name* | *node_id* , only the specified node is shut down; otherwise, the entire system is shut down.

Description

Use this command to shut down a single node or the entire clustered system in a controlled manner. You are prompted with a confirmation of intent (to process the command) when you specify this command.

If you enter this command with no parameters, the entire system is shut down. All data is flushed to disk before the power is removed.

If you enter this command with either a node ID or node name, the specified node is shut down. After the command completes, the remaining node in the I/O group enters write-through mode until the power to the node is returned, and the node rejoins the system.

Entering `y` or `Y` to the confirmation message processes the command. No feedback is then displayed. Entering anything other than `y` or `Y` results in the command not processing. No feedback is displayed.

If you need to shut down the entire system or a single node, use this command instead of using the power button on the nodes or powering off the main power supplies to the system.

Attention: Do not power off the uninterruptible power supply or remove the power cable from the node.

Storwize V7000: If you need to shut down the system or a single node, use this command instead of using the power button on power supplies, or powering off the mains to the system.

Before shutting down a node or system, complete the following requirements:

1. Quiesce all I/O operations that are destined for this node or system. If you do not quiesce these, failed I/O operations might be reported to your host operating systems.
2. Stop all FlashCopy, Metro Mirror, Global Mirror, and data migration operations.
3. Ensure that all asynchronous deletion operations have completed.

Using this command to shut down a single node fails if shutting down the node makes any volumes inaccessible, or if it is the last node in an I/O group. If you still need to shut down the node, you can use the **-force** option to override these checks.

Important: You can specify `stopsystem -node -reset` to restart the I/O process.

An invocation example

```
stopsystem
```

The following confirmation prompt is displayed:

```
Are you sure that you want to continue with the shut down?
```

Select `yes` to confirm or `no` to cancel.

swapnode

Use the **swapnode** command to exchange and maintain nodes without interruption to the virtualized target ports associated with the specified node.

Syntax

```
►►── swapnode ── ─┬──────────────┬─ ─┬───────┬─ ── -spare ─┬─ node_id ───┬─ ────────────────►
                  ├─ -replace ───┤   └─-force─┘             └─ node_name ─┘
                  ├─ -failback ──┤
                  ├─ -failover ──┤
                  └─ -service ───┘
```

```
►─┬─────────────┬──────────────────────────────────────────────────────────►◄
  ├─ node_id ───┤
  └─ node_name ─┘
```

Parameters

-replace | -failback | -failover | -service
(Optional) Specifies whether to replace or service a selected node. The values are:

- **-replace** replaces a specified offline node with a suitable candidate.
- **-failback** replaces a spare (that is being used) with the original node that it is replacing.
- **-failover** replaces a node with a spare even if it is currently online.
- **-service** puts a node into service state after the system triggers a failover because of any N_Port ID Virtualization (NPIV) ports.

-spare *node_id* | *node_name*
(Required) Specifies the ID of the spare node to use as a replacement. This parameter must be specified with **-failover**.

Note: The value for the ID must be greater than 1 because the spare is never the first node in the clustered system.

node_id | *node_name*
(Required) Specifies the node ID or name that is being swapped or serviced.

-force
Specifies that a node be removed, even if disruption to host system I/O might occur as a result.

Important: Specifying **-force** might result in a loss of access. Use it only under the direction of your product support information.

Description

This command exchanges and maintains nodes without interruption to the virtualized target ports associated with the specified node.

Specify **-replace** for the system to replace the name, I/O group, and site values associated with adding a node (by using the **addnode** command). These values are taken from the node that is being replaced. The existing node is explicitly specified, and a candidate node with the same WWNN value is chosen.

Remember: Specifying **-replace** might be used if you do not want to specify **rmnode** for an online node (which would delete the node from the clustered system).

An invocation example

```
swapnode -replace 2
```

The detailed resulting output:

```
No feedback
```

Chapter 9. Clustered system diagnostic and service-aid commands

Use the clustered system diagnostic and service-aid commands to diagnose and find clustered system (system) problems.

The system enables you to perform service activity, such as problem determination and repair activities, with a limited set of command-line tools. When you are logged in under the administrator role, all command-line activities are permitted. When you are logged in under the service role, only those commands that are required for service are enabled. The system diagnostic and service-aid commands apply under the service role.

applysoftware

Use the **applysoftware** command to update the clustered system (system) to a new level of system code (code).

Syntax

```
►►── applysoftware ──┬────────┬──────────────────────────────────────────────────►
                     └─ -force ┘

►──┬─ -file ── filename_arg ──┬────────┬──┬──────┬──┬───────────┬──┬──────────┬──┬──────►◄
   │                          └─ -pause ┘  └─ -all ┘  └─ -continue ┘  └─ -prepare ┘  │
   ├─ -abort ─────────────────────────────────────────────────────────────────────┤
   ├─ -complete ──┬──────────────────┬─────────────────────────────────────────────┤
   │              ├─ -paced ──────────┤                                             │
   │              └─ -delay ──minutes─┘                                             │
   ├─ -resume ──┬──────────────────┬───────────────────────────────────────────────┤
   │            ├─ -paced ──────────┤                                               │
   │            └─ -delay ──minutes─┘                                               │
   └─ -pacednext ──────────────────────────────────────────────────────────────────┘
```

Parameters

-force
(Optional) Specifies that the update or abort must proceed even if there is a lack of redundancy in the system. Disabling redundancy checking might cause loss of data, or loss of access to data. Use the **force** parameter with the **abort** parameter if one or more nodes are offline.

Important: Using the force parameter might result in a loss of access. Use it only under the direction of your product support information.

-file *filename_arg*
(Required) Specifies the file name of the installation update package. Copy the update package onto the configuration node before you run the **applysoftware** command.

Note: The **file** parameter cannot be used with the **abort** parameter.

-pause
(Optional) Specifies that the concurrent upgrade of a node is paused at the halfway point. This parameter must be specified with **-file**.

-all
(Optional) Specifies that the concurrent upgrade of a node is paused before the node is offline for an upgrade. This parameter must be specified with **-file**.

-continue
(Optional) Specifies that the concurrent upgrade continue.

-prepare
(Optional) Prepares the system for a manual code level update.

Note: You can:

- Use the **prepare** parameter with the **file** parameter
- Not use the **prepare** parameter with the **abort** parameter
- Not use the **force** parameter with the **prepare** parameter to go to prepared status

-abort
(Required for stopping an update) Specifies that a stalled or prepared update is stopped, returning the system to the original code level.

Note: The **abort** parameter can be used with the **force** parameter, but not the **file** or **prepare** parameters.

The **abort** parameter can also be used when the **lsupdate** command reports a status of:

- `prepare_failed`
- `prepared` (if all nodes are online)

-complete
(Required for completing an update) Specifies that the update completion process is started. Specify **-paced** for the update completion process to be paced. (This process is either automatic or paced.)

-resume
(Required for resuming an update) Resumes a stalled automatic update, update cancel, or update completion process by retrying the step that stalled. Specify **-paced** for the update process to be paced.

-paced
(Optional) Specifies that the update completion is paced. The system does not automatically take any nodes offline - you must specify **-pacednext** to indicate that the next node of the paced update is updated. Specify **-resume** for the update process to be resumed.

-pacednext
(Required if **-paced** is specified and you want the next node to be updated) Specifies that the next node that is part of a paced update be updated.

-delay *minutes*
(Optional) Specifies that customers can overwrite the default 30-minute delay at the half way point of a CCU.

Description

This command starts the update process of the system to a new level of code. The **applysoftware** command applies a level of code to the node as a service action (Paced update) to update the specific node, or as an automatic update process that update all of the nodes on a system.

The **applysoftware** command cannot be used in service state, which means the system must be running in order for the command to be used and be successful. This command is synchronous and therefore reports success or failure.

The code package as specified by the file name must first be copied onto the current configuration node in the /home/admin/update directory; use the PuTTy secure copy (scp) application to copy the file.

If the **applysoftware** command is successful, the **lsupdate** command reports the status is prepared. If the **applysoftware** command fails, the **lsupdate** command reports the status is inactive.

If specified, the **prepare** parameter must succeed in order to successfully update. It is recommended to use the same package for the prepare as the actual update. The **prepare** parameter can be canceled by using the abort parameter (even after the system is prepared) as long as the **lsupdate** command reports the status as prepared.

Important: The **prepare** parameter might time out. If a time-out occurs, the **prepare** parameter causes an asynchronous condition, and the **lsupdate** command reports the prepare status as preparing. If this happens, wait until **lsupdate** reports the update as prepared before you proceed with the manual update process.

The command completes as soon as the update process is successful. The command fails and the update package is deleted if:

- The specified package fails an integrity check due to corruption.
- Any node in the system has a hardware type that is not supported by the new code.
- The new code level does not support updates from the currently installed code.
- The code level of a remote system is incompatible with the new code.
- There are volumes-dependent on the status of a node.

  Note: The **force** parameter can be used to override these scenarios if you are prepared to lose access to data during the update. Before proceeding, use the **lsdependentvdisks** command with the **node** parameter to list the node-dependent volumes at the time the command is run. If the command returns an error, move the quorum disks to MDisks that are accessible through all nodes. Rerun the command until no errors are returned.

The actual update completes asynchronously.

An invocation example

```
applysoftware –file filename_arg
```

The resulting output:

```
No feedback
```

An invocation example

```
applysoftware -prepare -file INSTALL_6.4.0.0
```

The resulting output:

```
No feedback
```

An invocation example

```
applysoftware -abort
```

The resulting output:

```
No feedback
```

An invocation example

```
applysoftware -file softwareupdate
```

The resulting output:

```
No feedback
```

An invocation example

```
applysoftware -complete -force
```

The resulting output:

```
No feedback
```

An invocation example

```
applysoftware -resume -paced
```

The resulting output:

```
No feedback
```

An invocation example

```
applysoftware -pacednext -force
```

The resulting output:

```
No feedback
```

An invocation example

```
applysoftware -file jvardee1 -pause
```

The resulting output:

```
No feedback
```

An invocation example

```
applysoftware -file zibrav22 -all
```

The resulting output:

```
No feedback
```

An invocation example

```
applysoftware -continue
```

The resulting output:

```
No feedback
```

An invocation example

```
applysoftware -resume -delay 20
```

The resulting output:

```
No feedback
```

caterrlog (Deprecated)

The **caterrlog** command has been deprecated. Use the **lseventlog** command instead.

caterrlogbyseqnum (Deprecated)

The **caterrlogbyseqnum** command has been deprecated. Use the **lseventlog** command instead.

cherrstate (Deprecated)

The **cherrstate** command has been deprecated. Use the **cheventlog** command instead.

chdnsserver

Use the **chdnsserver** command to change a Domain Name System (DNS) server Internet Protocol (IP) address or name on a clustered system (system).

Syntax

```
►►── chdnsserver ── ┬──────────────────────┬ ┬────────────────────────┬ ──┬─ dns_name ─┬──────────►◄
                    └─ -ip ── ip_address ──┘ └─ -name ── dns_name ──┘   └─ dns_id ───┘
```

Parameters

-ip *ip_address*
(Optional) Specifies the DNS server IP address. The value must be in standard IPv4 or IPv6 format.

-name *DNS_name*
(Optional) Specifies a unique name for the system DNS server that is being changed.

dns_name | *dns_id*
(Required) Specifies DNS unique name or ID associated with the DNS server that is being changed.

Description

This command changes DNS server parameters (such as an IP address or name).

An invocation example

```
chdnsserver -ip 192.168.48.220 1
```

The resulting output:

```
No feedback
```

An invocation example

```
chdnsserver -name dns1 1
```

The resulting output:

```
No feedback
```

cheventlog

Use the **cheventlog** command to modify events in the event log.

Syntax

```
►►— cheventlog —-fix—sequence_number——————————————————————————————————►◄
```

Parameters

-fix *sequence_number*
(Required) Mark an unfixed event as fixed.

Description

Important: You must specify the **-fix** parameter.

An invocation example to mark an event fixed

```
cheventlog -fix 120
```

The resulting output:

```
No feedback
```

chsyslogserver

Use the **chsyslogserver** command to modify the parameters of an existing syslog server.

Syntax

```
►►— chsyslogserver — ┬—————————————————————————┬— ┬——————————————————————┬——————————————►
                     └ -name — server_name ┘       └ -ip — ip_address ┘

►┬————————————————————————┬— ┬—————————————————————┬— ┬———————————————————————┬— ——————————►
 └ -facility — facility ┘    └ -error ——┬ on ——┤      └ -warning ——┬ on ——┤
                                        └ off ┘                    └ off ┘

►┬————————————————————┬— ┬—————————————————————┬— —┬ syslog_server_name —┬——————————————►◄
 └ -info ——┬ on ——┤      └ -cadf ——┬ on ——┤       └ syslog_server_id ——┘
           └ off ┘                 └ off ┘
```

Parameters

-name *server_name*
(Optional) Specifies a name to assign to the syslog server. The name must be unique. When specifying a server name, syslog is a reserved word.

-ip *ip_address*
(Optional) Specifies an IP address to assign to the syslog server. This must be a valid IPv4 or IPv6 address.

-facility *facility*
(Optional) Specifies a facility number to identify the origin of the message to the receiving server.

Servers configured with facility values of `0 - 3` receive syslog messages in concise format. Servers configured with facility values of `4 - 7` receive syslog messages in fully-expanded format. This parameter is mutually-exclusive with **`-cadf`**.

`-error on | off`
(Optional) Specifies whether the server receives error notifications. Set to `on`, error notifications are sent to the syslog server. Set to `off`, error notifications are not sent to the syslog server.

`-warning on | off`
(Optional) Specifies whether the server receives warning notifications. Set to `on`, warning notifications are sent to the syslog server. Set to `off`, warning notifications are not sent to the syslog server.

`-info on | off`
(Optional) Specifies whether the server receives information notifications. Set to `on`, information notifications are sent to the syslog server. Set to `off`, information notifications are not sent to the syslog server.

`-cadf on | off`
(Optional) Specifies that Cloud Auditing Data Federation (CADF) data reporting be turned `on` or `off`. Any syslog notifications sent to the server are formatted to the CADF standard. This parameter is mutually-exclusive with **`-facility`**.

`syslog_server_name | syslog_server_id`
(Required) Specifies the name or ID of the server to be modified.

Description

Use this command to change the settings of an existing syslog server. You must specify either the current name of the server or the ID returned at creation time. Use the **`lssyslogserver`** command to obtain this ID.

If you disable CADF notifications for a syslog server that has CADF notification enabled, the facility value must be set to `0`.

An invocation example

```
chsyslogserver -facility 5 2
```

The resulting output:

```
No feedback
```

An invocation example

```
chsyslogserver -cadf on 0
```

The resulting output:

```
No feedback
```

clearerrlog

Use the **`clearerrlog`** command to clear all entries from the event log including status events and any unfixed errors.

Syntax

```
►►── clearerrlog ──┬──────────┬──────────────────────────────────────────►◄
                   └─ -force ─┘
```

Parameters

-force
(Optional) Specifies that the **clearerrlog** command be processed without confirmation requests. If the **-force** parameter is not supplied, you are prompted to confirm that you want to clear the log.

Description

This command clears all entries from the event log. The entries are cleared even if there are unfixed events in the log. It also clears any status events that are in the log.

Attention: This command is destructive. Use it only when you have either rebuilt the clustered system or have fixed a major problem that has caused entries in the event log that you do not want to manually fix.

An invocation example

```
clearerrlog -force
```

The resulting output:

```
No feedback
```

cpfabricdumps (Discontinued)

The **cpfabricdumps** command is discontinued. Use the **cpdumps** command instead.

dumperrlog

Use the **dumperrlog** command to dump the contents of the event log to a text file.

Syntax

```
►►── dumperrlog ──┬──────────────────────────────┬──────────────────────────►◄
                  └─ -prefix ── filename_prefix ─┘
```

Parameters

-prefix *filename_prefix*
(Optional) A file name is created from the prefix and a time stamp, and has the following format:

prefix_NNNNNN_YYMMDD_HHMMSS

where *NNNNNN* is the node front panel name.

Note: If the **-prefix** parameter is not supplied, the dump is directed to a file with a system-defined prefix of **errlog**.

Description

When run with no parameters, this command dumps the clustered system (system) event log to a file using a system-supplied prefix of `errlog`, which includes the node ID and time stamp. When a file name prefix is provided, the same operation is performed but the details are stored in the dumps directory within a file with a name that starts with the specified prefix.

A maximum of ten event-log dump files are kept on the system. When the 11th dump is made, the oldest existing dump file is overwritten.

Event log dump files are written to `/dumps/elogs`. The contents of this directory can be viewed using the **`lsdumps`** command.

Files are not deleted from other nodes until you issue the **`cleardumps`** command.

Note: The DMP family is printed at the end of all events in the event log.

An invocation example

```
dumperrlog -prefix testerrorlog
```

The resulting output:

```
No feedback
```

finderr

Use the **`finderr`** command to analyze the event log for the highest severity unfixed event.

Syntax

```
►►— finderr ————————————————————————————————————————————————►◄
```

Parameters

None

Description

The command scans the event log for any unfixed events. Given a priority ordering within the code, the highest priority unfixed event is returned to standard output.

You can use this command to determine the order in which to fix the logged event.

An invocation example

```
finderr
```

The resulting output

```
Highest priority unfixed event code is [1010]
```

setevent (Discontinued)

Attention: The **`setevent`** command is discontinued. SNMP notification can be configured using the following commands: **`svctask mksnmpserver`**, **`svctask chsnmpserver`**, **`svctask rmsnmpserver`**, and **`svcinfo lssnmpserver`**.

lscimomdumps (Deprecated)

The **`lscimomdumps`** command is deprecated. Use the **`lsdumps`** command to display a list of files in a particular dumps directory.

lscopystatus

Use the **`lscopystatus`** command to determine whether any file copies are currently in progress.

Syntax

```
►►— lscopystatus — ┬───────────┬─ ┬──────────────────────┬──────────────────────►◄
                   └─ -nohdr ──┘  └─ -delim — delimiter ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (**:**) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

Description

This command displays an indicator that shows if a file copy is in progress. Only one file can be copied in the clustered system at a time.

An invocation example

```
lscopystatus
```

The resulting output:

```
status
 active
```

lsdumps

Use the **lsdumps** command to display a list of files in a particular dumps directory on one of the nodes in the clustered system (system).

Syntax

```
►►— lsdumps — ┬───────────┬─ ┬───────────────────────┬─ ┬────────────────────────────┬──►
              └─ -nohdr ──┘  └─ -delim — delimiter ──┘  └─ -prefix — directory_name ─┘

►─ ┬───────────────┬──────────────────────────────────────────────────────────────►◄
   ├─ node_name ───┤
   └─ node_id ─────┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*

(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, then the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-prefix *directory_name*

(Optional) Specifies the name of the directory to list files for. The default is the `/dumps` directory. Valid directory names:

- `/dumps`
- `/dumps/audit`
- `/dumps/cimom`
- `/dumps/cloud`
- `/dumps/easytier`
- `/dumps/elogs`
- `/dumps/feature`
- `/dumps/iostats`
- `/dumps/iotrace`
- `/dumps/mdisk`
- `/home/admin/update`
- `/dumps/drive`
- `/dumps/enclosure`

node_name | *node_id*

(Optional) Specifies the node ID or name to list the available dumps for. If you do not specify a node, the available dumps on the configuration node are listed.

Description

This command displays a list of files that are detected by a node. You can specify the name of the directory to list files for, and the node ID or name. If you do not specify a directory, the `/dumps` directory is used.

The files are listed in order of time that is created, with the oldest files listed first.

Use the **lsdumps** command with the optional **prefix** parameter to specify a directory. If you do not specify a directory, `/dumps` is used as the default. Use the optional *node_id_or_name* parameter to specify the node to list the available dumps. If you do not specify a node, the available dumps on the configuration node are listed.

An invocation example to list the files in `/dumps` on the configuration node

```
lsdumps
```

The resulting output:

```
id   filename
0    svc.config.cron.bak_node1
1    svc.config.backup.xml_node1
2    recover.110584.100116.035201
3    dump.110584.100118.051550
4    ethernet.aaabbbX-1.trc
```

An invocation example to list the files in `/dumps/easytier` on the configuration node

```
lsdumps -prefix /dumps/easytier/ node_1
```

The resulting output:

```
id filename
0  dpa_heat.78RE5LV-1.150705.074636.data
1  dpa_log_78RE5LV-1_20150707062320_00000000.xml.gz
```

lsdnsserver

Use the **lsdnsserver** command to list information for any Domain Name System (DNS) servers in the clustered system (system).

Syntax

```
►►── lsdnsserver ── ─┬─────────────────────┬─ ─┬─────────┬─ ─┬────────────┬─ ──────────────►◄
                     └─ -delim ── delimiter ┘   └─ -nohdr ┘   ├─ dns_name ─┤
                                                              └─ dns_id ───┘
```

Parameters

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

dns_name | *dns_id*
(Optional) Specifies the DNS server name or ID for which to display details. The value for the ID must be a number and the value for the name must be an alphanumeric string.

Description

This command lists information for any DNS servers in the system.

This value provides the attribute values that can be displayed as output view data.

Table 54. lsdnsserver output

Attribute	Description
id	Indicates the DNS server ID. The value must be a number.
name	Indicates the DNS server name. The value must be an alphanumeric string.
type	Indicates the DNS server Internet Protocol (IP) address type. The value must be a standard IPv4 or IPv6 address.

Table 54. lsdnsserver output (continued)

Attribute	Description
IP_address	Indicates the IP address of the DNS server. The value must be a standard IPv4 or IPv6 address.

A concise invocation example

```
lsdnsserver
```

The resulting output:

```
id                    name                type                IP_address
0                     DNS1                ipv6                2801:0000:0000:0000:0000:0000:0000:0100
1                     DNS2                ipv4                192.168.44.34
```

A detailed invocation example

```
lsdnsserver 1
```

The resulting output:

```
id 1
name DNS2
type ipv4
IP_address 192.168.44.34
```

lserrlogbyfcconsistgrp (Deprecated)

The **lserrlogbyfcconsistgrp** command has been deprecated. Use the **lseventlog** command instead.

lserrlogbyfcmap (Deprecated)

The **lserrlogbyfcmap** command has been deprecated. Use the **lseventlog** command instead.

lserrlogbyhost (Deprecated)

The **lserrlogbyhost** command has been deprecated. Use the **lseventlog** command instead.

lserrlogbyiogrp (Deprecated)

The **lserrlogbyiogrp** command has been deprecated. Use the **lseventlog** command instead.

lserrlogbymdisk (Deprecated)

The **lserrorlogbymdisk** command has been deprecated. Use the **lseventlog** command instead.

lserrlogbymdiskgrp (Deprecated)

The **lserrlogbymdiskgrp** command has been deprecated. Use the **lseventlog** command instead.

lserrlogbynode (Deprecated)

The **lserrlogbynode** command has been deprecated. Use the **lseventlog** command instead.

lserrlogbyrcconsistgrp (Deprecated)

Attention: The **lserrlogbyrcconsistgrp** command has been deprecated. Use the **lseventlog** command instead.

lserrlogbyrcrelationship (Deprecated)

The **lserrlogbyrcrelationship** command has been deprecated. Use the **lseventlog** command instead.

lserrlogbyvdisk (Deprecated)

The **svcinfo lserrlogbyvdisk** command has been deprecated. Use the **svcinfo lseventlog** command instead.

lserrlogdumps (Deprecated)

The **lserrlogdumps** command is deprecated. Use the **lsdumps** command to display a list of files in a particular dumps directory.

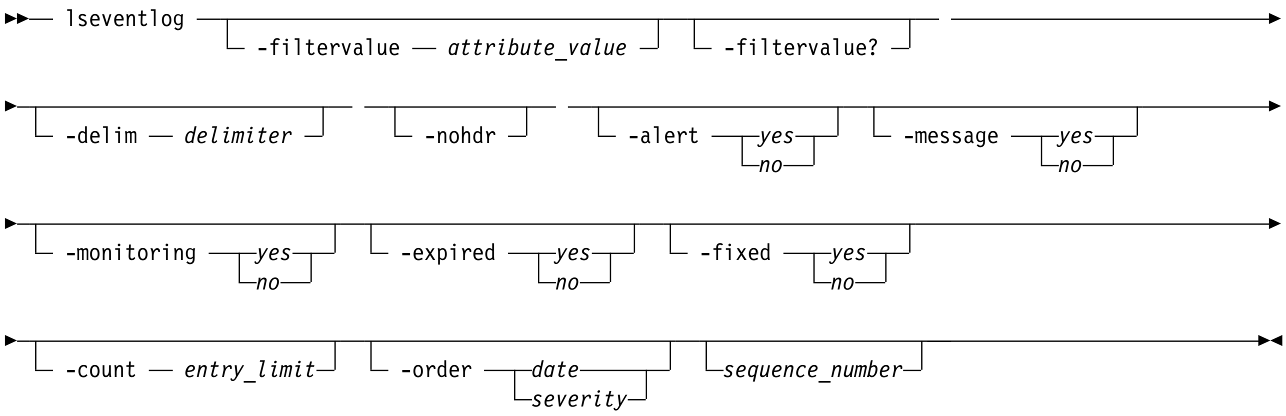
lsfeaturedumps (Deprecated)

The **lsfeaturedumps** command is deprecated. Use the **lsdumps** command to display a list of files in a particular dumps directory.

lseventlog

Use the **lseventlog** command to display a concise view of the system event log, or a detailed view of one entry from the log.

Syntax



Parameters

-filtervalue *attribute=value*

(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are returned. If a capacity is specified, the units must also be included.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards on the CLI:

- The wildcard character is an asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, you must enclose the filter entry within double quotation marks (""), as follows:

```
lseventlog -filtervalue "object_name=ob*"
```

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (**:**) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-alert *yes* | *no*
(Optional) Includes (or excludes) events with *alert* status.

-message *yes* | *no*
(Optional) Includes events with *message* status.

-monitoring *yes* | *no*
(Optional) Includes events with *monitoring* status.

-expired *yes* | *no*
(Optional) Includes (or excludes) events with *expired* status.

-fixed *yes* | *no*
(Optional) Includes (or excludes) events with *fixed* status.

-count *entry_limit*
(Optional) Indicates the maximum number of events to display.

-order *date* | *severity*
(Optional) Indicates what order the events must be in. Ordering by *date* displays the oldest events first. Ordering by severity displays the events with the highest severity first. If multiple events have the same severity, then they are ordered by date, with the oldest event displayed first.

The following list shows the order of severity, starting with the most severe:

1. Unfixed alerts (sorted by error code; the lowest error code has the highest severity)
2. Unfixed messages
3. Monitoring events (sorted by error code; the lowest error code has the highest severity)
4. Expired events
5. Fixed alerts and messages

-filtervalue?
(Optional) Displays a list of valid filter attributes for the **-filtervalue** *attribute=value* parameter:

- copy_id
- error_code
- event_count
- event_id
- fixed
- last_timestamp
- object_id
- object_name
- object_type

- report_node
- reporting_node_name
- root_sequence_number
- sequence_number status

sequence_number
(Optional) Indicates whether the command must display a full view of the event.

Description

This command displays a concise view of the system event log, or a detailed view of one entry from the log. You can sort the events and entries by severity or age.

The default values for included events are:

- alert=yes
- expired=no
- fixed=no
- message=yes
- monitoring=no

Table 55 provides the attribute values that can be displayed as output view data.

Table 55. `lseventlog` output

Attribute	Description	Value
machine_type	Indicates the node machine type and model number.	The value is an alphanumeric string up to 7 characters long.
serial number	Indicates the node serial number.	The value is an alphanumeric string up to 7 characters long.
sequence_number	Indicates the sequence number of the event.	The value is numeric from `0` to `8000000`.
first_timestamp	Indicates when the event was added to the log.	The value is in the format `YYMMDDHHMMSS`.
first_timestamp_epoch	Indicates when the event was added to the log (in seconds) after the epoch occurs.	The value is a numeric 32-bit value.
last_timestamp	Indicates when the event was most recently updated.	The value is in the format `YYMMDDHHMMSS`.
last_timestamp_epoch	Indicates the most recent update (in seconds) after an epoch for an event.	The value is a numeric 32-bit value.
fixed_timestamp	Indicates the time stamp when event is fixed.	The value is in the format `YYMMDDHHMMSS`.
fixed_timestamp_epoch	Indicates the time stamp (in seconds) when an event is fixed after an epoch occurs.	The value is a numeric string.
fru	Indicates the field-replaceable unit (FRU) for error or event; this field contains probable FRUs (separated by commas).	The value is an ASCII string up to 255 characters long.

Table 55. **lseventlog** *output (continued)*

Attribute	Description	Value
object_type	Indicates the type of the object the event is logged against.	The values are: • mdisk • mdiskgrp • volume (or volume copy) • node • host • io_grp (iogroup in **dumperrlog**) • fc_consist_grp (fcgrpin **dumperrlog**) • rc_consist_grp(rcgrp in **dumperrlog**) • fc_map (fcmap in **dumperrlog**; *flash* in **caterrlog**) • rc_relationship (rcmapin **dumperrlog**; *remote* in **caterrlog**) • cluster • controller (devicein **caterrlog** and **dumperrlog**) • quorum • migrate • email_server (*emailserver* in **caterrlog** and **dumperrlog**) • enclosure • drive
object_id	Indicates the ID of the object the event is logged against.	The value is a numeric 64-bit value. It is displayed in decimal for all object types other than clustered systems. For a clustered system, this value is hexadecimal but is blank for events with cluster object types.
object_name	Indicates the name of the object the event is logged against.	This value is the object name format and is blank if the object was deleted or does not have a name.
copy_id	Indicates the volume copy ID the event is logged against.	The value is a numeric value 0 to 1; it is blank if not a vdiskcopy event.
reporting_node_id	Indicates the ID of the node that reported the event.	The value is a numeric 64-bit value that is blank if the event is reported by the clustered system.
reporting_node_name	Indicates the name of the node that reported the event.	This value is the object name format and is blank if node is deleted or event is reported by the clustered system.
root_sequence_number	Indicates the sequence number of the root or causal event **Important:** If the event is directly caused by another event, the sequence_number of the related event is shown here.	The value is a numeric value from 1 to 8000000; blank if there is no root or if the event is not directly caused by another event.

Table 55. `lseventlog` output (continued)

Attribute	Description	Value
event_count	Indicates the number of reported events that are combined into this event	The value is a numeric 32-bit value.
status	Indicates the event category.	The values are: • `alert` • `message` • `monitoring` • `expired`
fixed	Indicates whether the event was marked fixed (for an alert) or read (for a message).	The values are: • `yes` • `no` (for events that cannot be fixed, or are not fixed)
auto_fixed	Indicates whether event is marked fixed by the code.	The values are: • `yes` • `no` (for events that cannot be fixed, or are not fixed)
notification_type	Indicates the type of event notification.	The values are: • `error` • `warning` • `informational` • `none`
event_id	Indicates the event ID.	The value is a 6-digit numeric value.
event_id_text	Indicates the description that is associated with the event ID.	The value is a text value with a maximum of 200 bytes. This appears in CLI requested language.
error_code	Indicates the error code that is associated with this event.	The value is a 4-digit numeric value but is blank if there is no error code.
error_code_text	Indicates the description that is associated with the error code.	The value is a text value with a maximum of 200 bytes that is blank if there is no error code. This value appears in the language that is requested by the CLI.
description	Indicates the description that is associated with the event. If the event has an error code, this value is the same as the `error_code_text` field; otherwise, it is the same as the `event_id_text` field	Text (maximum of 200 bytes).

Table 55. `lseventlog` *output (continued)*

Attribute	Description	Value
sense1	Indicates the sixteen bytes of hex-encoded sense data; least significant byte is on the left.	The value is 16 2-character hexadecimal numbers that are separated by spaces.
sense2		
sense3		
sense4		
sense5		
sense6		
sense7		
sense8		

Invocation examples

This example shows events in January 2010:

```
lseventlog -filtervalue 'last_timestamp>=100101000000:last_timestamp<100201000000'
```

This example shows all unfixed 1065 errors, in order of occurrence:

```
lseventlog -filtervalue error_code=1065:fixed=no
```

This example lists the most critical event:

```
lseventlog -order severity -count 1
```

This example shows the concise view:

```
lseventlog

sequence_number:last_timestamp:object_type:object_id:object_name:copy_id:
status:fixed:event_id:error_code:description

400:100106132413:vdisk:2:my_vdisk:1:alert:no:060001:1865:
Space Efficient Virtual Disk Copy offline due to insufficient space
401:100106140000:cluster::ldcluster-2::message:no:981001:
:Cluster Fabric View updated by fabric discovery
```

This example shows the full view:

```
lseventlog 120

sequence_number 120
first_timestamp 111130100419
first_timestamp_epoch 1322647459
last_timestamp 111130100419
last_timestamp_epoch 1322647459
object_type node
object_id 1
object_name node1
copy_id
reporting_node_id 1
reporting_node_name node1
root_sequence_number
event_count 1
status alert
fixed yes
auto_fixed no
notification_type error
event_id 073003
event_id_text More/Less fibre channel ports operational
error_code 1060
error_code_text Fibre Channel ports not operational
```

```
machine_type 21458F4
serial_number 75BZPMA
fru none
fixed_timestamp 111202141004
fixed_timestamp_epoch 1322835004


sense1 03 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00
sense2 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
sense3 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
sense4 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
sense5 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
sense6 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
sense7 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
sense8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

lssyslogserver

Use the **lssyslogserver** command to return a concise list or a detailed view of syslog servers that are configured on the clustered system.

Syntax

```
►►── lssyslogserver ── ─┬─────────┬─ ─┬──────────────────────┬─ ─┬────────────────────┬──────►◄
                        └─ -nohdr ┘   └─ -delim ── delimiter ┘   ├─ syslog_server_name ┤
                                                                 └─ syslog_server_id ──┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum possible width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a one-byte character. If you enter -delim : on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

syslog_server_name | *syslog_server_id*
(Optional) Specifies the name or ID of an existing syslog server. When you use this parameter, a detailed view of the specified syslog server is returned. If you do not specify a syslog server name or ID, then a concise view of all syslog servers is displayed.

Description

Use this command to display a concise list or a detailed view of syslog servers that are configured on the clustered system.

This table provides the attribute values that can be displayed as output view data.

Table 56. lssyslogserver output

Attribute	**Description**
id	Indicates the syslog server ID. The value must be a number.
name	Indicates the syslog server name. The value must be an alphanumeric string.
IP_address	Indicates the syslog server Internet Protocol (IP) address. The value must be a valid IP address.
facility	Indicates the syslog server facility value. The value must be a number 0 - 7 but is blank for a CADF notification-enabled server.
error	Indicates whether error messages are on. The values are `on` or `off`.
warning	Indicates whether warning messages are on. The values are `on` or `off`.
info	Indicates whether informational messages are on. The values are `on` or `off`.
cadf	Indicates whether CADF data reporting is enabled or not for the syslog server. The values are `on` or `off`.

A concise invocation example

```
lssyslogserver -delim :
```

The concise resulting output:

```
id:name:IP_address:facility:error:warning:info:cadf
0:syslog0:192.135.60.4::on:on:on:on
1:newserver:192.136.70.7:4:on:off:off:on
```

A detailed invocation example

```
lssyslogserver 0
```

The detailed resulting output:

```
id 0
name syslog0
IP_address 192.135.60.4
facility
error on
warning on
info on
cadf on
```

lssoftwaredumps (Deprecated)

The **lssoftwaredumps** command is deprecated. Use the **lsdumps** command to display a list of files in a particular dumps directory.

lssoftwareupgradestatus (Deprecated)

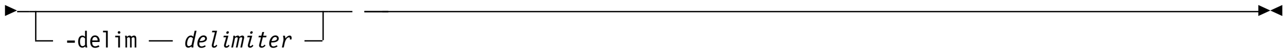
The **lssoftwareupgradestatus** command is deprecated. Use the **lsupdate** command instead.

lssystemsupportcenter

Use the **lssystemsupportcenter** command to list details about remote support servers.

Syntax

```
►►── lssystemsupportcenter ── ──────────────────────── ── ─────────── ── ─────────────────────────────►
                              └─support_center_name─┘    └─ -nohdr ─┘
                              └─support_center_id───┘
```



Parameters

support_center_name | *support_center_id*
(Optional) Specifies a name or ID for a remote support server in the server index. The value for the ID must be a number (integer) and the value for the name must be an alphanumeric string. This parameter displays a full view of any configured name or ID values.

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If no data is displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

Description

This command lists details about remote support servers.

This command returns a concise or detailed view of the remote support servers defined.

This table provides the attribute values that can be displayed as output view data.

Table 57. lssystemsupportcenter output

Attribute	Description
id	Indicates the support center or proxy server ID. The value must be a number (integer).
name	Indicates the support center or proxy server name. The value must be an alphanumeric string.
IP_address	Indicates the Internet Protocol Version 4 (IPv4) or Version 6 (IPv6) address for the new support center or proxy server. The value must be a valid IPv4 or IPv6 address.
port	Indicates the port number for the configured support center or the proxy server. The value must be a number (integer).
proxy	Indicates that the target server is a proxy server (and not the support center). The values are `yes` or `no`.

An invocation example

```
lssystemsupportcenter
```

The following output is displayed:

```
id name IP_address port proxy
0 proxy1 1.2.3.4 9999 yes
1 supportserver2 1.2.3.5 8888 no
2 test_frontend_server 9.51.88.165 1025 no
```

An invocation example

```
lssystemsupportcenter 2
```

The following output is displayed:

```
id 2
name supportserver2
IP_address 1.2.3.5
port 8888
proxy no
```

An invocation example

```
lssystemsupportcenter -delim :
```

The following output is displayed:

```
id:name:IP_address:port:proxy
0:proxy1:1.2.3.4:9999:yes
1:supportserver2:1.2.3.5:8888:no
2:test_frontend_server:9.51.88.165:1025:no
```

lsupdate

Use the **lsupdate** command to display a system's machine code (code) upgrade status.

Syntax

```
►►── lsupdate ── ─┬────────┬─ ─┬───────────────────┬──────────────────────────►◄
                  └─ -nohdr ┘   └─ -delim ── delimiter ┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim : on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

Description

The following are the upgrade status states:

status Indicates the overall update-related status of the system. The values are:

- success, which indicates that all updating activity is complete.
- system_preparing, which indicates that the system is preparing a manual update.
- system_prepared, which indicates that the system is ready to starting a manual update.
- system_prepare_failed, which indicates that the system could not start a manual update.

  Note: Check the event log.

- system_initializing, which indicates that the system is readying nodes for an update.
- system_updating, which indicates that the nodes are being updated.
- system_updating_pausing, which indicates that the system is pausing before it continues to update the nodes.

  Note: If you specify applysoftware -continue the status changes to system_updating.
- system_committing, which indicates that all nodes are updated and the system is readying the new code.
- system_stalled, which indicates that an update is stalled because of unexpected node problems.
- system_stalled_non_redundant, which indicates that an update is stalled because of dependent volumes.
- system_restoring, which indicates that a stalled update is canceled by the user.

  Note: The previous code version is being restored.
- system_restoring_pausing, which indicates that the system is pausing before it continues to restore the nodes.

  Note: If you specify applysoftware -continue the status changes to system_updating_restoring.
- system_restoring_stalled_non_redundant, which indicates that an update is canceled and then stalled because of dependent volumes.
- system_manual_update, which indicates that a manual update is in progress.
- system_completion_required, which indicates that all nodes are updated and management functions are available, but further system changes are necessary to complete the update.

  Note: Check the event log.
- system_completing, which indicates that an automatic update completion is in progress.
- system_completing_pausing, which indicates that automatic update completion is paused.
- system_completing_paced, which indicates that a paced update completion is in progress.
- system_completing_stalled, which indicates that an automatic update completion stalled because of an unexpected problem.
- enclosures, which indicates that enclosure firmware is being updated.
- enclosures_stalled, which indicates that an enclosure firmware update stalled because of an unexpected problem or the enclosure has a lack of redundancy.
- 1 enclosures_restoring, which indicates a stalled update has been canceled by the user. The
  1 previous code version is being restored to the enclosure canisters.
- drives, which indicates that drive firmware is being updated.

event_sequence_number
Indicates an event that describes any current problem with the code update. The value must be a numeric string in decimal format (or blank).

progress
Indicates the completion percentage of the current update activity in terms of number of objects updated (rather than time elapsed). The value must be a numeric string (decimal) in the range 0 - 100.

estimated_completion_time
Indicates estimated completion time of current update activity. It is valid only if the current update activity is automatic and is not stalled. The value must be in the format YYMMDDHHMMSS (or blank).

suggested_action
Indicates the actions that help the update progress. The value must be:

- complete, which indicates the system update is complete and update completion must be issued. Nodes are online.
- continue, which indicates that the concurrent upgrade is paused and you must applysoftware -continue to continue the concurrent upgrade.
- fix, which indicates that an update cannot continue because of a problem. Check the event log, specifically the **event_sequence_number** output. Nodes are offline.
- manual, which indicates that a manual update is in progress.
- pacednext, which indicates that the paced update is in progress and the next node must be scheduled for updating. Nodes are online.
- resume_cancel, which indicates that the update is stalled but can be resumed or canceled. Nodes are online.
- resume, which indicates that the update completion is stalled but can be resumed. Nodes are online.
- start, which indicates that the system is ready for a new update to start. No update is in progress or prepared and all nodes are online.
- wait, which indicates that the system is busy (no action is required) because an update is in progress.

system_new_code_level
Indicates that a new level of code is being updated. The value must be the build version (or blank if not updating or restoring the system).

system_forced
Indicates any current node-related activity in forced mode (ignoring dependent volumes). The values are yes or no.

system_next_node_status
Indicates the status of the next node in the current node-related update activity. The values are:

- none, which indicates that there is no node to update.
- paused, which indicates that the current node is paused during a concurrent upgrade, and you must applysoftware -continue to continue the concurrent upgrade.
- waiting, which indicates that the node is ready for updating and that the system is waiting (typically for multipathing failover).
- ready, which indicates that the node is ready for updating, and the update activity is paced. You must start the update manually.
- updating, which indicates that the node is updating.
- stalled, which indicates that the node is going to be updated next, but the update is stalled.

system_next_node_time
Indicates the time that the next node update will start. It is valid only if the **system_next_node_status** is waiting. The value must be in the format YYMMDDHHMMSS x (or blank).

system_next_node_id
Indicates the ID of the next node in the current node-related update. The value must be a numeric string (or blank).

system_next_node_name
Indicates the name of the next node in the current node-related update. The value must be an alphanumeric string (or blank).

An invocation example of an update

```
lsupdate
```

The resulting output:

```
status system_updating
event_sequence_number
progress 50
estimated_completion_time 140522093020
suggested_action wait
system_new_code_level 7.4.0.1 (build 99.2.141022001)
system_forced no
system_next_node_status updating
system_next_node_time
system_next_node_id 2
system_next_node_name node2
```

An invocation example of a paced update

```
lsupdate
```

The resulting output:

```
status system_completing_paced
event_sequence_number
progress 75
estimated_completion_time
suggested_action pacednext
system_new_code_level
system_forced no
system_next_node_status ready
system_next_node_time
system_next_node_id 4
system_next_node_name node4
```

mkdnsserver

Use the **mkdnsserver** command to configure a new Domain Name System (DNS) server for a clustered system (system).

Syntax

```
►►— mkdnsserver — — -ip — ip_address — ┬────────────────────┬— ————————————————————►◄
                                       └ -name — DNS_name ┘
```

Parameters

-ip *ip_address*
(Required) Specifies the DNS server Internet Protocol (IP) address. The value must be in standard IPv4 or IPv6 format. Depending on the format you specify the system will validate the format to make sure it is correct.

-name *DNS_name*
(Optional) Specifies a unique name for the system DNS server being created. If a DNS server name is not specified, a unique name is generated and then assigned to the DNS server.

Description

This command configures a new DNS server for a system.

Use the **-ip** parameter to specify the DNS server IP address. You can enter any valid IPv4 or IPv6 address. The system validates the format of specified IP address to make sure that it is correct.

An invocation example

```
mkdnsserver -ip 192.168.44.34
```

The resulting output:

```
DNS Server id [0] successfully created
```

An invocation example

```
mkdnsserver -ip 2801:0000:0000:0000:0000:0000:0000:0100
```

The resulting output:

```
DNS Server id [1] successfully created
```

mksyslogserver

Use the **mksyslogserver** command to create a syslog server to receive notifications.

Syntax

```
►►── mksyslogserver ── ──────────────────────────── ── -ip ── ip_address ──────────────────────────►
                        └─ -name ── server_name ─┘

►─────────────────────────────── ────────────────────────── ───────────────────────────── ───────────────►
 └─ -facility ── facility ─┘     │            ┌─ on ──┐  │   │              ┌─ on ──┐  │
                                 └─ -error ───┴─ off ─┴──┘   └─ -warning ───┴─ off ─┴──┘

►───────────────────────── ────────────────────────────────────────────────────────────────────────►◄
 │           ┌─ on ──┐  │   └─ -cadf ──┬─ on ──┬─┘
 └─ -info ───┴─ off ─┴──┘              └─ off ─┘
```

Parameters

-name *server_name*
(Optional) Specifies a unique name to assign to the syslog server. If a name is not specified, then a system default of syslog*n* is applied, where *n* is the ID of the server. When specifying a server name, syslog is a reserved word.

-ip *ip_address*
(Required) Specifies the Internet Protocol (IP) address of the syslog server. This must be a valid Internet Protocol Version 4 (IPv4) or Internet Protocol Version 6 (IPv6) address.

-facility *facility*
(Optional) Specifies the facility number used in syslog messages. This number identifies the origin of the message to the receiving server. Servers configured with facility values of 0 - 3 receive syslog messages in concise format. Servers configured with facility values of 4 - 7 receive syslog messages in fully-expanded format. The default value is 0.

-error on | off
(Optional) Specifies whether the server receives error notifications. Set to on, error notifications are sent to the syslog server. Set to off, error notifications are not sent to the syslog server. The default value is on.

-warning on | off
(Optional) Specifies whether the server receives warning notifications. Set to on, warning notifications are sent to the syslog server. Set to off, warning notifications are not sent to the syslog server. The default value is on.

-info on | off
(Optional) Specifies whether the server receives information notifications. Set to `on`, information notifications are sent to the syslog server. Set to `off`, information notifications are not sent to the syslog server. The default value is `on`.

-cadf on | off
(Optional) Specifies that Cloud Auditing Data Federation (CADF) data reporting be turned `on` or `off`. Any syslog notifications sent to the server are formatted to the CADF standard. This parameter is mutually-exclusive with **-facility**.

Description

This command creates a syslog server to receive notifications. The syslog protocol is a client-server standard for forwarding log messages from a sender to a receiver on an IP network. Syslog can be used to integrate log messages from different types of systems into a central repository.

SAN Volume Controller supports a maximum of six syslog servers.

An invocation example

```
mksyslogserver -ip 1.2.3.4
```

The resulting output:

```
Syslog Server id [2] successfully created
```

An invocation example

```
mksyslogserver -ip 9.193.231.37 -error on -warning on -info off -cadf on
```

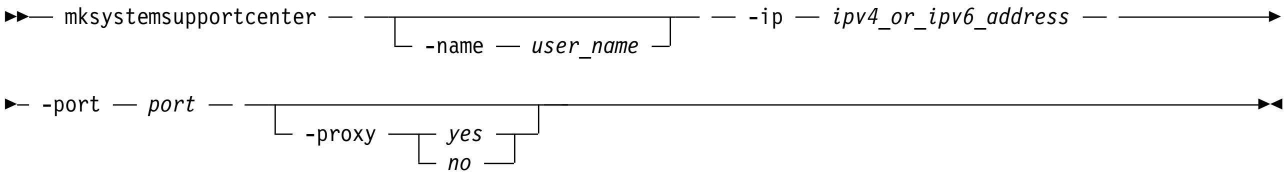
The resulting output:

```
Syslog Server id [2] successfully created
```

mksystemsupportcenter

Use the **mksystemsupportcenter** command to add a support center or proxy server to your remote support configuration.

Syntax



Parameters

-name *user_name*
(Optional) Specifies the unique name of the support center or proxy to be defined. If a center with that name is defined, the command fails. The value must be an alphanumeric string that:

- Cannot start with the string `default_support_center`
- Cannot start with a hyphen or number
- Cannot begin or end with a space
- Must be 1 - 64 characters long (using numbers, letters, spaces, periods, or an underscore)

Additionally, the names SupportCenter and Proxy cannot be used as shown. Do not specify these names unless you use all lower-case letters - for example, supportcenter or proxy.

Note: If you do not specify a name, a system default of supportservern is used, where n is the object index.

-ip *ipv4_or_ipv6_address*
(Required) Indicates the Internet Protocol Version 4 (IPv4) or Version 6 (IPv6) address for the new support center or proxy server. The value must be a valid IPv4 or IPv6 address.

-port *port*
(Required) Specifies the port number for the new connection. The value must be a number in the range 1 - 65535. The default value is 1025.

-proxy *yes* | *no*
(Optional) Specifies whether a target server is a proxy server (and not a support center). The values are yes or no.

Description

This command creates a support center or proxy server to your remote support configuration. The maximum number of proxy servers that you can define is six.

If you configure a proxy server, remote support assistance is available only by using proxy server (which means that no direct connections are tried). If you configure multiple support centers (or proxies), the system cycles through each support center before it enables remote support assistance which means that it tries all support centers at least three times before it ends in failure state. Use the set of default support centers that are configured if needed.

Important: Unless your support team recommends it, do not configure any new support centers. However, a proxy server can be configured if required. If you are routing Remote Support connections of a system with all the nodes configured with a service IPv6 address, then you must use a Remote Support proxy server. The proxy server should listen on an IPv6 address. It can additionally listen on IPv4 address. This proxy server IPv6 address needs to be specified on your system using this command (with the -proxy parameter).

An invocation example

```
mksystemsupportcenter -name test_frontend_server_1 -ip 9.51.88.165 -port 1025
```

The resulting output:

```
Support Server id [0] successfully created
```

An invocation example

```
mksystemsupportcenter -name customer_proxy_1 -ip 192.168.56.88 -port 9999 -proxy yes
```

The resulting output:

```
Support Server id [1] successfully created
```

An invocation example

```
mksystemsupportcenter -name customer_proxy_2 -ip 192.168.56.101 -port 2222 -proxy yes
```

The resulting output:

```
Support Server id [2] successfully created
```

An invocation example

```
mksystemsupportcenter -ip 2001:0db8:0000:0001:0000:0000:0000:0071 -port 1025 -proxy yes
```

The resulting output:

```
 Support Server id [3] successfully created
```

An invocation example

```
mksystemsupportcenter -ip 2001:db8:0:1:0:0:0:71 -port 1025 -proxy yes
```

The resulting output:

```
 Support Server id [2] successfully created
```

An invocation example

```
mksystemsupportcenter -ip 2001:db8:0:1::71 -port 1025 -proxy yes
```

The resulting output:

```
 Support Server id [2] successfully created
```

rmdnsserver

Use the **rmdnsserver** command to remove a Domain Name System (DNS) server from a clustered system (system).

Syntax

```
►►— rmdnsserver — ─┬─ dns_name ─┬─ ──────────────────────────────────────────►◄
                   └─ dns_id ───┘
```

Parameters

dns_name | *dns_id*
: (Required) Specifies the ID or name of the Domain Name System (DNS) server to be removed from the system. The value must be a number for the ID and an alphanumeric string for the name.

Description

This command removes a DNS server from a system.

An invocation example

```
rmdnsserver 1
```

The resulting output:

```
No feedback
```

rmsyslogserver

Use the **rmsyslogserver** command to delete the specified syslog server.

Syntax

```
►►— rmsyslogserver — ─┬─ syslog_server_name ─┬────────────────────────────►◄
                      └─ syslog_server_id ───┘
```

Parameters

syslog_server_name | *syslog_server_id*
(Required) Specifies the name or ID of the syslog server to be deleted.

Description

Use this command to delete an existing syslog server. You must specify either the current name of the server or the ID returned at creation time. Use the **lssyslogserver** command to obtain this ID.

An invocation example

```
rmsyslogserver 2
```

The resulting output (if the command is successful):

```
No feedback
```

If the command is not successful, this error message occurs:

```
CMMVC5753E The specified object does not exist or is not a suitable candidate.
```

rmsystemsupportcenter

Use the **rmsystemsupportcenter** command to delete a configured support center or proxy server.

Syntax

```
►►── rmsystemsupportcenter ── ┬─system_support_name─┬──────────────────────────────────────►◄
                              └─system_support_id───┘
```

Parameters

system_support_name | *system_support_id*
(Required) Specifies a name or ID for a remote support server in the server index that is to be removed. The value for the ID must be a number (integer) and the value for the name must be an alphanumeric string. The name or ID must correspond to the entry that is displayed when you specify **lssystemsupportcenter**.

Description

This command deletes a configured support center or proxy server.

Note: You cannot remove default support centers that are automatically configured for the system.

An invocation example

```
rmsystemsupportcenter 0
```

The resulting output:

An invocation example

```
rmsystemsupportcenter secret_proxy_server
```

The resulting output:

```
No feedback
```

An invocation example

```
rmsystemsupportcenter special_support_centre
```

The resulting output:

```
No feedback
```

setlocale

Use the **setlocale** command to change the locale setting for the clustered system (system). It also changes command output to the chosen language.

Syntax

```
▶▶— setlocale — — -locale — locale_id ————————————————————————————————————————▶◀
```

Parameters

-locale *locale_id*
Specifies the locale ID. The value must be a numeric value depending on the desired language (as indicated below)

Description

This command changes the language in which error messages are displayed as output from the command-line interface. Subsequently, all error messages from the command-line tools are generated in the chosen language. This command is run when you request a change of language (locale).

Specify the **setlocale** command to change the locale setting for the system; all interface output is changed to the chosen language. For example, to change the language to Japanese, type the following:

```
setlocale -locale 3
```

where 3 is the value for Japanese. The following values are supported:

- 0 US English (default)
- 1 Simplified Chinese
- 2 Traditional Chinese
- 3 Japanese
- 4 French
- 5 German
- 6 Italian
- 7 Spanish
- 8 Korean
- 9 Portuguese (Brazilian)
- 10 Russian

An invocation example (where 3 is Japanese)

```
setlocale -locale 3
```

The resulting output:

```
No feedback
```

An invocation example (where 8 is Korean)

```
setlocale -locale 8
```

The resulting output:

```
No feedback
```

svqueryclock

Use the **svqueryclock** command to return the date, time, and current time-zone of the clustered system (system).

Syntax

```
►►—svqueryclock——————————————————————————————————————————————————————►◄
```

Parameters

None

Description

This command returns the date, time and current time-zone of the system.

An invocation example

```
svqueryclock
```

The resulting output:

```
Mon Nov 25 14:59:28 GMT 2013
```

writesernum

Use the **writesernum** command to write the node serial number into the planar NVRAM.

Syntax

```
►►— writesernum — — -sernum — serial_number — ┬ node_id ──┬──────────────────►◄
                                              └ node_name ┘
```

Parameters

-sernum *serial_number*
(Required) Specifies the serial number to write to the nonvolatile memory of the system planar.

node_id | *node_name*
(Required) Specifies the node where the system planar is located. The serial number is written to this system planar. This name is not the worldwide node name (WWNN).

Description

This command writes the node serial number into the planar NVRAM and then reboots the system. You can find the serial number at the front of the node without having to remove it from the rack. The seven-digit alphanumeric serial number is located on a label on the front of the node. The serial number on the label might contain a hyphen. Omit this hyphen when typing the serial number with the **writesernum** command.

Note: Once you have written the serial number to the planar NVRAM, you can issue the **lsnodevpd** command to verify that the number is correct. The `system_serial_number` field contains the serial number.

An invocation example

```
writesernum -sernum 1300027 node1
```

The resulting output:

```
No feedback
```

Chapter 10. Controller commands

Use the controller commands to modify the name of a storage controller.

chcontroller

Use the **chcontroller** command to modify the attributes of a controller.

Syntax

```
►►── chcontroller ── ──┬───────────────────────┬── ──┬────────────────────────────┬──────────────────►
                       └─ -name ── new_name ───┘     └─ -allowquorum ──┬─yes─┬────┘
                                                                        └─no──┘

►──┬──────────────────────────────┬──┬─ controller_id ───┬────────────────────────────────────────►◄
   ├─ -site ──┬─ site_name ──┬────┤  └─ controller_name ─┘
   │          └─ site_id ────┘    │
   └─ -nosite ────────────────────┘
```

Parameters

-name *new_name*
(Optional) Specifies the new name to be assigned to the controller.

-allowquorum yes | no
(Optional) Specifies that the controller is allowed or is not allowed to support quorum disks. A value of yes enables a suitable controller to support quorum disks. A value of no disables a controller from supporting quorum disks, provided that the specified controller is not currently hosting a quorum disk.

-site *site_name* | *site_id*
(Optional) Specifies the numeric site value or site name for the controller. The value is 1, 2, or 3.

Note: The controller site cannot be changed if the system topology is stretched or hyperswap (and there are MDisks).

-nosite
(Optional) Resets the site value for the controller.

controller_id | *controller_name*
(Required) Specifies the controller to modify. Use either the controller name or the controller ID.

Description

This command changes the name of the controller that is specified by the *controller_id* | *controller_name* variable to the value that you specify with the **-name** parameter.

If any controller that is associated with an MDisk shows the **allow_quorum** attribute set to no with the **lscontroller** command, the set quorum action fails for that MDisk. Before using the **chcontroller** command to set the **-allowquorum** parameter to yes on any disk controller, check the following website to see whether the controller supports quorum.

www.ibm.com/support

You can add a new disk controller system to your SAN at any time. Follow the switch zoning guidelines in the section about switch zoning. Also, ensure that the controller is set up correctly for use with the clustered system (system).

To add a new disk controller system to a running configuration, ensure that the system has detected the new storage MDisks by issuing the **detectmdisk** command. The controller has automatically been assigned a default name. If you are unsure of which controller is presenting the MDisks, issue the **lscontroller** command to list the controllers. The new controller is listed with the highest numbered default name. Record the controller name and follow the instructions in the section about determining a disk controller system name.

Give this controller a descriptive name by issuing the following command:

```
chcontroller -name newname oldname
```

List the unmanaged MDisks by issuing the following command:

```
lsmdisk -filtervalue mode=unmanaged:controller_name=newname
```

These MDisks correspond to the RAID arrays or partitions that you have created. Record the field controller LUN number. The field controller LUN number corresponds with the LUN number that you assigned to each of the arrays or partitions.

Create a new storage pool and add only the RAID arrays that belong to the new controller to this storage pool. Avoid mixing RAID types; for each set of RAID array types (for example, RAID-1 or RAID-10), create a new storage pool. (You cannot use RAID-10 with distributed arrays.) Assign this storage pool an appropriate name; if your controller is called `FAST650-abc` and the storage pool contains RAID-10 arrays, assign the MDisk a name similar to `F600-abc-R5`. Issue the following command:

```
mkmdiskgrp -ext 16 -name mdisk_grp_name
-mdisk colon-separated list of RAID-x mdisks returned
```

Note: This creates a new storage pool with an extent size of 16 MB.

An invocation example

```
chcontroller -name newtwo 2
```

The resulting output:

```
No feedback
```

An invocation example

```
chcontroller -site site1 controller18
```

The resulting output:

```
No feedback
```

lscontroller

Use the **lscontroller** command to display a concise list or a detailed view of controllers that are visible to the clustered system (system).

Syntax

```
►►── lscontroller ──┬──────────────────────────────────┬──┬───────────┬──────────────►
                    └─ -filtervalue ── attribute=value ─┘  └─ -nohdr ──┘
```

```
►──┬────────────────────┬── ── ──┬───────────────┬──┬─ controller_id ───┬───────────────────────►◄
   └─ -delim ── delimiter ┘        └─ -filtervalue? ┘  └─ controller_name ─┘
```

Parameters

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are returned. If a capacity is specified, the units must also be included.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards with the CLI:

- The wildcard character is an asterisk (`*`).
- The command can contain a maximum of one wildcard, which must be the first or last character in the string.

-filtervalue?
(Optional) Displays the valid filter attributes. The following filter attributes for the **lscontroller** command are valid:

- controller_name
- id
- site_id
- site_name

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

controller_id | *controller_name*
(Optional) Specifies the name or ID of a controller. When you use this parameter, the detailed view of the specific controller is returned and any value that is specified by the **-filtervalue** parameter is ignored. If you do not specify the *controller_id* | *controller_name* parameter, the concise view displays all controllers that match the filtering requirements that are specified by the **-filtervalue** parameter.

Description

This command returns a concise list, or a detailed view, of controllers visible to the system.

The following values are applicable to the data in the output views:

```
degraded no, yes
```

To differentiate the name of a storage controller from the name that is shown on the system, list the storage controllers by issuing the **lscontroller** command. Record the controller name or ID for the controller that you want to determine. For the controller in question, issue the **lscontroller** *controller* *name* | *id* command, where *controller name* | *id* is the controller name or ID. Record the worldwide node name (WWNN) for the controller. You can use the WWNN to determine the actual storage controller by launching the native controller user interface, or by using the command line tools that it provides to verify the actual controller that has the WWNN.

Notes:

1. The *mdisk_link_count* value is the number of MDisks currently associated with this storage controller.
2. The *max_mdisk_link_count* value is the highest value that the *mdisk_link_count* reaches since it was last reset to the *mdisk_link_count* value.

   Remember: This value is reset by specific maintenance procedures or when the event log is cleared.
3. A SAN connection from a node or node canister port to a controller port for a single MDisk is a path. The controller port *path_count* value is the number of paths that are currently being used to submit input/output (I/O) data to this controller port.
4. The storage controller *max_path_count* value is the highest value that the storage controller *path_count* reaches since it was last reset to the *path_count* value. This value is reset by specific maintenance procedures or when the system error log is cleared.

   Important: The *max_path_count* value is the highest value that the *path_count* reaches since it was last reset to the *path_count* value.

   Remember: This value is reset by specific maintenance procedures or when the event log is cleared.
5. The *allow_quorum* value identifies if the controller is enabled to support quorum disks. Quorum support is either enabled or disabled depending on the controller hardware type.
6. The *ctrl_s/n* value is the controller serial number.

   Important: This data comes from vendor-controlled sources and might not be available.

Table 58 provides the attribute values that can be displayed as output view data.

This table provides the attribute values that can be displayed as output view data.

Table 58. lscontroller output

Attribute	**Possible Values**
id	Indicates the controller ID.
name	Indicates the controller name.
WWNN	Indicates the worldwide node name (WWNN). This field is blank for iSCSI controllers.
mdisk_link_count	Indicates the MDisk link count.
max_mdisk_link_count	Indicates the maximum MDisk link count.
degraded	Indicates whether the controller has degraded MDisks.
vendor_id	Indicates the vendor identification name or number.
product_id_low	Indicates the product identification.
product_id_high	Indicates the product identification.
product_revision	Indicates the product revision.
ctrl_s/n	Indicates the controller serial number.
allow_quorum	Indicates that the controller can support quorum disks.
WWPN	Indicates the worldwide port name (WWPN). This field is blank for iSCSI controllers.

Table 58. lscontroller output (continued)

Attribute	Possible Values
path_count	Indicates the number of paths that are currently being used to submit input/output (I/O) data to the controller port.
max_path_count	Indicates the maximum number of paths that are currently being used to submit input/output (I/O) data to the controller port.
site_id	Indicates the site value for the controller. This numeric value is 1, 2, 3, or blank.
site_name	Indicates the site name for the controller. This value is an alphanumeric value or is blank.
fabric_type	Indicates a Fibre Channel (FC) or SAS controller. • `fc` indicates an FC controller • `sas_direct` indicates an SAS direct-attached controller • `multiple` indicates multiple controllers (either FC, SAS, or both) • `iscsi` indicates an iSCSI controller
iscsi_port_id	Indicates the I/O port identifier, which is the same as the WWPN value from the FC domain. This value shows the iSCSI port ID for the iSCSI controller and is blank for other controllers. This value must be a numeric value. This ID refers to the row number in output from the **lsiscsistorageport** command. The **lsiscsistorageport** can be used to find the controller IQN.
ip	Indicates the IP address that is associated with the `iscsi_port_id`. This value shows the IP value for the iSCSI controller and is blank for other controllers. This value must be an IPv4 or IPv6 address.
physical_capacity	Indicates the physical capacity of the controller. This value is always blank or empty for controllers that do not report physical capacity information.

A concise invocation example for an iSCSI controller

```
lscontroller -delim :
```

The concise resulting output:

```
id:controller_name:ctrl_s/n:vendor_id:product_id_low:product_id_high:WWNN:degraded:fabric_type:site_id:site_name
0:controller0::IBM:1726-4xx:FAStT::no:iscsi:1:snpp1
1:controller1::IBM:1726-4xx:FAStT::no:iscsi:2:snpp2
7:controller7:3EK0J5Y8:SEAGATE :ST373405:FC:200600A0B851061E:yes:fc:1:snpp1
8:controller8:3EK0J6CR:SEAGATE :ST373405:FC:200600A0B851061D:no:fc:2:snpp2
9:controller9:3EK0J4YN:SEAGATE :ST373405:FC:200600A0B851061C:no:fc:3:snpp3
10:controller10:3EK0GKGH:SEAGATE :ST373405:FC:200600A0B851061B:no:fc:1:snpp4
11:controller11:3EK0J85C:SEAGATE :ST373405:FC:200600A0B851061A:no:fc:2:snpp5
12:controller12:3EK0JBR2:SEAGATE :ST373405:FC:200600A0B851062A:no:fc:3:snpp6
13:controller13:3EKYNJF8:SEAGATE :ST373405:FC:200600A0B851062B:no:fc:1:snpp7
14:controller14:3EK0HVTM:SEAGATE :ST373405:FC:200600A0B851062C:no:fc:2:snpp8
```

A detailed invocation example for a Fibre Channel controller

```
lscontroller -delim = 7
```

The detailed resulting output:

```
id=7
controller_name=controller7
WWNN=20000004CF2412AC
mdisk_link_count=1
max_mdisk_link_count=1
degraded=no
vendor_id=SEAGATE
product_id_low=ST373405
product_id_high=FC
product_revision=0003
ctrl_s/n=3EK0J5Y8
allow_quorum=no
site_id=2
site_name=DR
WWPN=22000004CF2412AC
path_count=1
max_path_count=1
WWPN=21000004CF2412AC
path_count=0
max_path_count=0
fabric_type=sas_direct
iscsi_port_id=
ip=
physical_capacity=20.0GB
```

A detailed invocation example for an iSCSI controller

```
lscontroller 0
```

The detailed resulting output:

```
id 0
controller_name controller0
WWNN
mdisk_link_count 4
max_mdisk_link_count 4
degraded no
vendor_id IBM
product_id_low 1726-4xx
product_id_high   FAStT
product_revision 0617
ctrl_s/n
allow_quorum no
fabric_type iscsi
site_id
site_name
WWPN
path_count 4
max_path_count 4
iscsi_port_id 1
ip 10.10.10.1
WWPN
path_count 4
max_path_count 4
iscsi_port_id 2
ip 10.10.10.2
physical_capacity 40.0GB
```

lscontrollerdependentvdisks

Use the **lscontrollerdependentvdisks** command to list the volumes that depend on the specified controller.

Syntax

```
►►── lscontrollerdependentvdisks ── ┬─────────┬─ ┬────────────────────┬─ ─────────────────►
                                    └─ -nohdr ┘  └─ -delim ── delimiter ┘

►─┬─ controller_id_list ───┬───────────────────────────────────────────────────────►◄
  └─ controller_name_list ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim : on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

controller_id_list | *controller_name_list*
Specifies one or more controller IDs, controller names, or both. Separate multiple controllers using the colon character (:).

Description

The **lscontrollerdependentvdisks** command lists the volumes that depend on the status of the specified controllers. If a controller goes offline, the dependent volumes also go offline. Before you take a controller offline for maintenance, you can use the command to ensure that you do not lose access to any volumes.

If you have multiple controllers that are configured as a single subsystem, you must specify all of the controllers in the subsystem. When you do this you must specify a single command invocation.

The **lscontrollerdependentvdisks** command also checks for quorum disks on the specified controller list. If any quorum disks are on the specified controller list, the command returns an error. All quorum disks must be moved before you perform any maintenance. After you move quorum disks, reissue the command to list the dependent volumes.

Note: The command lists the volumes that depend on the controllers at the time the command is run; subsequent changes to your system require rerunning the command.

An invocation example

```
lscontrollerdependentvdisks controller0
```

The concise resulting output:

```
vdisk_id vdisk_name
0 vdisk0
1 vdisk1
2 vdisk2
```

Chapter 11. Drive commands

Use the drive commands to capture information to assist with managing drives.

applydrivesoftware

Use the **applydrivesoftware** command to update drives.

Syntax

```
►►── applydrivesoftware ── -file ──name──┬──────────────────────────┬──┬─ -drive ──drive_id─┬────────────►
                                         │            ┌─firmware─┐  │  └─ -all ─────────────┘
                                         └─ -type ────┴─fpga─────┴──┘

►─┬──────────┬──┬───────────────────┬──┬───────────────────┬─────────────────────────────────────────►◄
  └─ -force ─┘  └─ -allowreinstall ─┘  └─ -allowdowngrade ─┘


►►── applydrivesoftware ── -cancel ──────────────────────────────────────────────────────────────────►◄
```

Parameters

-file *name*
(Required) Specifies the firmware update file name that must be copied to the /home/admin/update/ directory on the configuration node.

-type fpga | firmware
(Optional) Specifies the type of drive firmware to update. Drive firmware updates can be performed online, concurrently with I/O. However, fpga updates require the drive to be taken offline, which means target drives must be made candidate before issuing the **applydrivesoftware** command. The default value is firmware. See the **chdrive** command for more details.

-all
(Optional) Specifies that the drive firmware should be applied to every drive in the system, as long as that drive is online and has use member, use spare, or use candidate.

This does not apply to:

- Drives that have dependent volumes
- Drives that are members of non-redundant arrays

Drives hosting quorum qualify, but there is risk. To avoid this risk use **-drive** and make sure the quorum is moved in between **applydrivesoftware** invocations. Use the **chquorum** command to avoid updating a drive that is hosting quorum.

If you specify **-all** you must specify the **-type** as firmware.

Remember: The **-all** parameter differs from the **-drive** parameter because unsuitable drives are not added to the list of drives scheduled for update when you use **-all**.

-drive *drive_id*
(Optional) Specifies one drive ID or a list of drive IDs (separated by a colon, [**:**]) to be updated. The maximum number of IDs is 128. If you have more than 128, use **-all** or multiple **applydrivesoftware** invocations to complete the update.

Remember: The **-drive** parameter differs from the **-all** parameter because if you specify an unsuitable drive using the **-drive** parameter, the **applydrivesoftware** command fails. Additionally, if you specify all three drives as hosting quorum, the command fails. If you use the **-drive** option to specify a single drive, and that drive has use=unused, it is updated.

-force
(Optional) Bypasses the dependent volume check. By default **applydrivesoftware** cannot run if any volumes dependent on the drive. Specifying **-force** bypasses this check, allowing the drive software update to proceed. Drive-dependent volumes generally result from non-redundant or degraded RAID arrays.

Note: Restore redundancy to the system (where possible) instead of using the **-force** parameter.

Important: Using the **-force** parameter might result in a data loss. Use it only under the direction of your product support information, or if you are willing to accept the risk of data loss in the array or pool to which the drive belongs.

-allowreinstall
(Optional) Specifies to make the system install the current level (again) onto drives that contain a file in the package.

Remember: Using this parameter is not recommended.

-allowdowngrade
(Optional) Specifies to allow the system to downgrade the firmware on a drive (that contains a file in the package).

Remember: Use this parameter only under the direction of your product support information.

-cancel
(Optional) Specifies that the command be stopped.

Description

Use this command to update the firmware of drives that are managed by the system.

There are two types of drive software that can be updated using this command:

- firmware
- fpga

Drive firmware updates can be performed online while the drive is in use. When used on an array member drive **applydrivesoftware** checks for volumes that are dependent on the drive and refuses to run if any are found. Drive dependent volumes are usually caused by non-redundant or degraded RAID arrays. Where possible you should restore redundancy to the system by replacing any failed drives before using the **applydrivesoftware** command. When this is not possible, for example on drives that are members of a RAID-0 array, you can either add redundancy to the volume by adding a second copy in another pool, or use the **-force** parameter to bypass the dependant volume check.

Remember: Only use **-force** if you are willing to accept the risk of data loss on dependent volumes (if the drive fails during the firmware update).

Drive firmware updates occur asynchronously, and conclude after the **applydrivesoftware** command completes. To see the status of the updates, use the **lsdriveupgradeprogress** command.

Drive fpga updates might require the drive to be taken offline for several minutes. Drives must be changed to the candidate state before **applydrivesoftware** can be used to update fpga software. The fpga updates occur asynchronously, continuing in the background after the **applydrivesoftware** command has returned. You must check the FPGA_level field in **lsdrive** *N*, where *N* is the *drive_id*, to see whether or not the update completed successfully.

Remember: Interrupting an fpga update by removing power from the drive or enclosure might make the drive unusable. Only one drive's fpga can be updated per **applydrivesoftware** invocation. Make sure that the update is complete before unseating the drive or removing power from the enclosure.

An invocation example

```
applydrivesoftware -file DRIVE_XXXXXXXX -type firmware -drive 4
```

The resulting output:

```
No feedback
```

An invocation example

An example that fails because of drive-dependent volumes:

```
applydrivesoftware -file DRIVE_XXXXXXXX -type firmware -drive 6
```

The resulting output:

```
CMMVC6953E The action cannot be completed because vdisks are dependent on the specified mdisk.
Force is required.
```

An invocation example

```
applydrivesoftware -file drivemicrocodepackagev5 -type firmware -all
```

The resulting output:

```
No feedback
```

An invocation example

```
applydrivesoftware -file drivemicrocodepackagev1 -type firmware -all -allowreinstall
```

The resulting output:

```
No feedback
```

An invocation example

```
applydrivesoftware -file drivemicrocodepackagev1 -type firmware -all -allowdowngrade
```

The resulting output:

```
No feedback
```

An invocation example

```
applydrivesoftware -file drivemicrocodepackagev1 -type firmware -all -allowdowngrade
-allowreinstall
```

The resulting output:

```
No feedback
```

An invocation example

```
applydrivesoftware -cancel
```

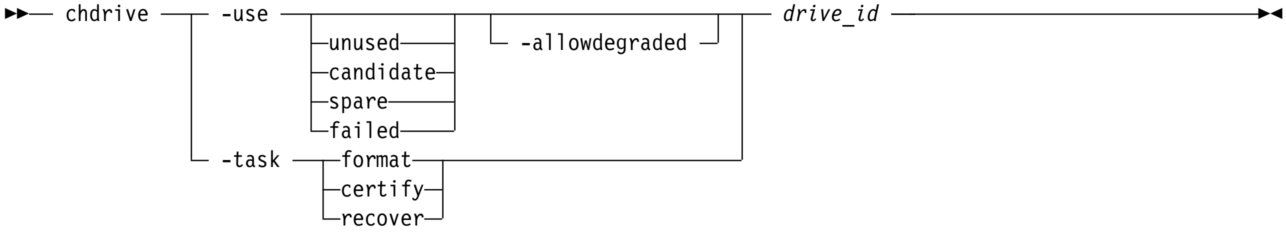
The resulting output:

```
No feedback
```

chdrive

Use the **chdrive** command to change the drive properties.

Syntax



Parameters

-use unused | candidate | spare | failed
Describes the role of the drive:

- unused indicates the drive is not in use and will not be used as a spare
- candidate indicates the drive is available for use in an array
- spare indicates the drive can be used as a hot-spare drive if required
- failed indicates the drive has failed.

Note: To create member drives, add the drives to (new) arrays using the **mkarray** command.

If a drive fails for a distributed array, the array remains associated with the failed drive while it is in the failed state.

-allowdegraded
(Optional) Permits permission for a change of drive use to continue, even if a hotspare drive is not available for the array that the drive is a member of. You cannot specify **-allowdegraded** and **-task** together.

Important: Using **-allowdegraded** is not recommended.

-task format | certify | recover
Causes the drive to perform a task:

- format indicates a drive will be formatted for use in an array; only permitted when drive is a candidate or has failed validation.
- certify indicates the drive will be analyzed to verify the integrity of the data it contains; permitted for any drive that is a member.
- recover recovers an offline flash drive without losing data; permitted when the drive is offline because a build is required, or when the drive has failed validation.

drive_id
The identity of the drive.

Description

Use this command to change the drive role, or to start long running drive tasks.

You can use **lsdriveprogress** to display progress (percentage) and estimated completion time of ongoing drive tasks.

When a drive associated with a distributed array is changed from `member` to `failed`, if the distributed array does not have available rebuild space then it is degraded. If **-allowdegraded** is not specified the command fails because of insufficient rebuild areas . If the **-allowdegraded** parameter is specified the command succeeds and the array no longer uses the drive for I/O operations. If a drive is changed from `failed` to another configuration the distributed array is forgets about the drive and creates a missing member that belongs in the member table. Use the **charraymember** command to replace the missing member.

An invocation example

```
chdrive -use spare 1
```

The resulting output:

```
No feedback
```

An invocation example to certify drive 23

```
chdrive -task certify 23
```

The resulting output:

```
No feedback
```

lsdrive

Use the **lsdrive** command to display configuration information and drive vital product data (VPD).

Syntax

```
►►— lsdrive — ┬──────────────────────────────────┬ ┬─────────┬ ┬──────────────────────┬ ┬─────────────────┬ ┬────────┬ ┬──────────┬ ──────────►◄
              └ -filtervalue — attribute_value ┘ └ -nohdr ┘ └ -delim — delimiter ┘ └ -filtervalue? ┘ └-bytes┘ └drive_id┘
```

Parameters

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards:

- The wildcard character is the asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, enclose the filter entry within double quotation marks (""): `lsdrive -filtervalue mdisk_id="1*"`

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (**:**) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-filtervalue?
(Optional) Displays the valid filter attributes for the **-filtervalue** parameter:

- capacity
- enclosure_id
- error_sequence_number
- id
- interface_speed
- mdisk_id
- mdisk_name
- member_id
- node_id
- node_name
- slot_id
- status
- tech_type
- use
- drive_class_id

-bytes
(Optional) The size (capacity) of the drive in bytes.

drive_id
(Optional) The identity of the drive.

Description

Use this command to display configuration information and drive VPD.

Note: Filtering should be permitted on all concise fields.

Table 59 on page 365 describes possible outputs.

*Table 59. **lsdrive** output*

Attribute	Value
id	Indicates the ID of the drive: • `online`, which indicates that the drive is available through all drive ports. • `degraded`, which indicates that the drive is available but not through all drive ports. • `offline`, which indicates that the drive is unavailable.
status	Indicates the summary status of the drive.
error_sequence_number	Indicates the error sequence number that describes the cause of the drive status: • `online`, which is blank. • `degraded`, which is populated if associated with an error. • `offline`, which must be populated. **Note:** Error sequence numbers indicate an entry in the event log. This value includes entries that are both errors, and informational messages (for example, the drive is formatting).
use	Indicates the current role of the drive: • `unused` if the drive that is not configured to be used by anything. • `candidate` if the drive is available to be configured. • `spare` if the drive is configured as a spare, to be used if the arrays fail members. • `member` if the drive is configured as a member of an array. • `failed` if the drive is rejected and is no longer available for use.
UID	Indicates that the unique ID reported by the drive.
tech_type	Indicates the drive technology used. The values are: • `unsupported` indicates that the drive is not supported in this platform (contact your support team). • tier0_flash • tier1_flash • tier_enterprise • tier_nearline
replacement_date	Indicates the date of a potential drive failure. The format must be `YYMMDD`. This value is blank for non-SSD drives.
capacity	Indicates the capacity of disk, excluding quorum area.
block_size	Indicates the block size of the disk.
vendor_id	Indicates the manufacturer of the drive.
product_id	Indicates the product ID of the drive.
FRU_part_number	Indicates the FRU part number of the drive.
FRU_identity	Indicates the 11S number that combines manufacturing part number and serial number.
RPM	Indicates the specified RPM of the disk.
firmware_level	Indicates the firmware level of the disk; blank if unknown.
FPGA_level	Indicates the FPGA level, if applicable; blank if not applicable or unknown.
mdisk_id	Indicates the ID of the array MDisk that the drive is a member of.
mdisk_name	Indicates the name of the MDisk that the drive is a member of.
member_id	Indicates the ID of the MDisk array member.

*Table 59. **lsdrive** output (continued)*

Attribute	**Value**
enclosure_id	Indicates whether the: • Drive is contained in an enclosure (not a node) and the slot position is known, this value is the ID of the enclosure in which the drive is located. • Drive is contained in a node (not an enclosure), this value is blank. • Enclosure ID is not determined yet, this value is blank.
slot_id	Indicates the slot_id of the drive in the enclosure or node. It can be referred to as the drive bay or location. This value can be blank.
node_name	Indicates the node name where the drive is located for a drive that is contained within a node. For a drive contained within an enclosure, it is blank.
node_id	Indicates the node ID where the drive is located for a drive that is contained within a node. For a drive contained within an enclosure, blank.
quorum_id	Indicates the ID of quorum disk; blank if not quorum disk.
port_1_status	Indicates the connectivity status of the drive's first port. The values are `online`, `offline`, or `excluded`. **Note:** Port 1 is attached to the node that has a panel name that ends in `-1`.
port_2_status	Indicates the connectivity status of the drive's second port. The values are `online`, `offline`, or `excluded`. **Note:** Port 2 is attached to the node that has a panel name that ends in `-2`.
interface_speed	Indicates the lowest interface speed for the connected drive slot (in gigabits per second, or Gbps). The values are: • 1.5 Gbps • 3 Gbps • 6 Gbps • 12 Gbps • Blank if both ports are isolated or the drive is not connected
protection_enabled	Indicates whether SCSI type-2 protection information is enabled (`yes`) or not (`no`).
auto_manage	Indicates whether the `auto_manage` process is running (`active`) or not running (`idle`).
drive_class_id	Indicates which drive class the drive is part of.
write_endurance_used	Indicates the drive writes per day (DWPD). This value is blank for drives that are not SSD drives. The value must be a number 0 - 255. This value indicates the percentage of life that is used by the drive. The value `0` indicates that full life remains, and `100` indicates that the drive is at or past its end of life. **Note:** The drive must be replaced when the value exceeds `100`. This value is blank for drives that are either: 1. Not SSDs. 2. SSDs that predate support of the endurance indicator. This value also applies to drives that are yet to be polled, which can take up to 24 hours.
write_endurance_usage_rate	Indicates the DWPD usage rate. The values are: • `measuring` • `high` • `marginal` • `low` This value is blank for non-SSD drives. **Note:** This field displays a value only when the `write_endurance_used` value changes.

A concise invocation example

```
lsdrive -delim :
```

The concise resulting output:

```
id:status:error_sequence_number:use:tech_type:capacity:mdisk_id:mdisk_name:member_id:enclosure_id:slot_id:auto_manage:dr
0:online::member:tier0_flash:20GB:0:mdisk0:0:1:2:active:0
1:offline:345:member:tier0_flash:20GB:0:mdisk0:0:1:3:idle:0
2:online::member:tier0_flash:20GB:0:mdisk0:0:1:4:active:0
```

A detailed invocation example for an SSD drive

```
lsdrive 0
```

The detailed resulting output:

```
id:0
status:online
error_sequence_number:
use:member
UID:20000004cf4cd2c0

tech_type:tier0_flash
capacity:20GB
block_size:512
vendor_id:IBM
product_id:I8MR1337 W00Y4Y1
FRU_part_number:AAAAAAA
FRU_identity:11S1817115Y41337171001
RPM:
firmware_level:3.02
FPGA_level:1.99
mdisk_id:0
mdisk_name:mdisk0
member_id:0
enclosure_id:1
slot:2
node_id:
node_name:
quorum_id:
port_1_status:online
port_2_status:online
interface_speed:6Gb
protection_enabled:yes
auto_manage:active
drive_class_id:3
write_endurance_used:5
write_endurance_usage_rate:high
work_load:high
replacement_date:190806
```

A detailed invocation example for a tier 1 flash SSD drive

```
lsdrive 0
```

The detailed resulting output:

```
id 0
status degraded
error_sequence_number
use candidate
UID 5000c5002624a723
tech_type sas_hdd
capacity 1.8TB
block_size 512
vendor_id IBM-207x
product_id ST32000444SS
FRU_part_number 85Y5869
```

```
FRU_identity 11S41Y8471YXXX9WM40LMD
RPM 10000
firmware_level BC2D
FPGA_level
mdisk_id
mdisk_name
member_id
enclosure_id 1
slot_id 7
node_id
node_name
quorum_id 0
port_1_status online
port_2_status offline
interface_speed 6Gb
protection_enabled no
auto_manage inactive
drive_class_id 3
write_endurance_used 30
drive_class_id
write_endurance_used 5
write_endurance_usage_rate high
work_load high
replacement_date 190806
```

A detailed invocation example for a hard disk drive (HDD)

```
lsdrive 0
```

The detailed resulting output:

```
id 0
status degraded
error_sequence_number
use candidate
UID 5000c5002624a723
tech_type sas_nearline_hdd
capacity 1.8TB
block_size 512
vendor_id IBM-207x
product_id ST32000444SS
FRU_part_number 85Y5869
FRU_identity 11S41Y8471YXXX9WM40LMD
RPM 7200
firmware_level BC2D
FPGA_level
mdisk_id
mdisk_name
member_id
enclosure_id 1
slot_id 7
node_id
node_name
quorum_id 0
port_1_status online
port_2_status offline
interface_speed 6Gb
protection_enabled no
auto_manage inactive
drive_class_id 3
write_endurance_used
drive_class_id
write_endurance_used 5
write_endurance_usage_rate high
work_load high
replacement_date 190806
```

lsdriveclass

Use the **lsdriveclass** command to display all drive classes in the clustered system (system).

Syntax

```
►►— lsdriveclass — ┬───────┬─ ┬───────────────┬─ ───────────────────────────────────►
                   └ -nohdr┘  └ -filtervalue? ┘

►┬──────────────────────────────────┬─ ┬──────────────────────┬─ ┬──────────────────┬──►◄
 └ -filtervalue — attribute=value ──┘  └ -delim — delimiter ──┘  └ drive_class_id ──┘
```

Parameters

-nohdr
: (Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-filtervalue *attribute=value*
: (Optional) Specifies a list of one or more filter attributes that match the specified values; see **-filtervalue?** for the supported attributes.

Note: Some filters allow the use of a wildcard when you specify the command. The following rules apply to the use of wildcards when you use the CLI:

- The wildcard character is an asterisk (*).
- The command can contain a maximum of one wildcard, which must be the first or last character in the string.
- When you specify a wildcard character, you must enclose the filter entry within double quotation marks (""), as follows:

```
lsdriveclass -filtervalue  "IO_group_name=md*"
```

-filtervalue?
: (Optional) Includes all of the valid filter attributes in the report. The following filter attributes are valid for the **lsdriveclass** command:

- id
- RPM
- capacity
- IO_group_id
- IO_group_name
- tech_type
- block_size
- candidate_count
- superior_count
- total_count

Any parameters that are specified with the **-filtervalue?** parameter are ignored.

-delim *delimiter*
: (Optional) By default, in a concise view all columns of data are space-separated, with the width of each column set to the maximum width of each item of data. In a detailed view, each item of data is an individual row, and if you display headers, the data is separated from the header by a space. The

-delim parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. Specify `-delim :` on the command line, and the colon character (`:`) separates all items of data in a concise view (for example, the spacing of columns does not occur); in a detailed view, the specified *delimiter* separates the data from its header.

drive_class_id
(Optional) The identity of the drive class.

Description

This command displays all drive classes in a system. Drives are displayed if they are managed.

Table 60 provides the attribute values that can be displayed as output view data.

Table 60. `lsdriveclass` output

Attribute	Possible Values
id	Indicates the drive class ID.
RPM	Indicates the speed of the drive class.
capacity	Indicates the capacity of the drive class.
IO_group_id	Indicates the I/O group ID associated with the drive class.
IO_group_name	Indicates the I/O group name that is associated with the drive class.
tech_type	Indicates the technology type of the drive class.
block_size	Indicates the block size of the drive class.
candidate_count	Indicates the number of drives in the drive class that are in `candidate` state.
superior_count	Indicates the total number of drives in this class and the drives that count as superior. This value applies to distributed arrays created by using **mkdistributedarray**.
total_count	Indicates the total number of drives in the drive class. The drive state is irrelevant.

A concise invocation example

```
lsdriveclass -filtervalue block_size=4096
```

The detailed resulting output:

```
 id  RPM    capacity   IO_group_id  IO_group_name  tech_type  block_size  candidate_count  superior_count  total_count
 3   15000  600.5GB    2            io_group2      tier0_flash    4096        0                0               24
```

A concise invocation example

```
lsdriveclass -filtervalue io_group_ID=0:tech_type=tier_enterprise
```

The detailed resulting output:

```
id  RPM    capacity   IO_group_id  IO_group_name  tech_type  block_size  candidate_count  superior_count total_count
0   10000  300.9GB    0            io_group0      tier0_flash    512         30               3              30
```

A concise invocation example

```
lsdriveclass -delim ! -nohdr
```

The detailed resulting output:

```
0!10000!300.9GB!0!io_group0!tier_nearline!512!30!30
1!!600.5GB!0!io_group0!tier_nearline!512!10!50
2!15000!900.1GB!1!io_group1!tier_enterprise!512!60!60
3!15000!600.5GB!2!io_group2!tier_enterprise!4096!0!24
```

A detailed invocation example

```
lsdriveclass 2
```

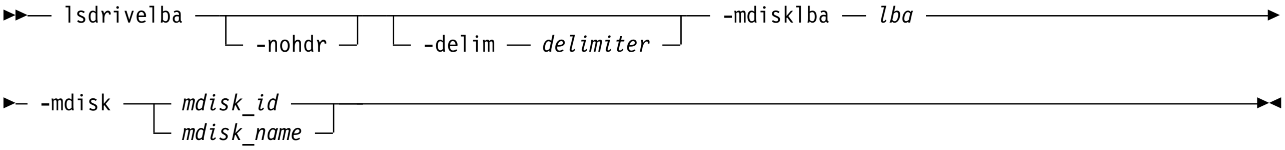
The detailed resulting output:

```
id 2
RPM 15000
capacity 900.1GB
IO_group_id 1
IO_group_name io_group1
tech_type tier0_flash
block_size 512
candidate_count 60
superior_count 5
total_count 60
```

lsdrivelba

Use the **lsdrivelba** command to map array MDisk logical block address (LBA) to a set of drives.

Syntax



Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim : on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-mdisklba *lba*
(Required) The logical block address (LBA) on the MDisk. The LBA must be specified in hex, with a 0x prefix.

-mdisk*mdisk_id* | *mdisk_name*
(Required) The ID or name of the MDisk.

Description

This command maps the array MDisk LBA to a set of drives.

The system provides volumes that have LBAs for 512-byte block sizes, but back-end disks that have a block size of either 512 or 4096 bytes can also be used. Drives are listed in their physical size.

Use the **lsdrive** command to display the drive block size, and use the **lsdrive** or `lsarray` command to list each object (the drive and the MDisk).

Table 61 describes possible outputs.

Table 61. lsdrivelba output

serial-attached SCSI | **| | (SAS)Attribute**	**Value**
drive_id	The ID of drive; blank if no configured array member exists (for example, in a degraded array).
type	The type of information on the disk: • `parity`, in which LBA range contains `parity` (RAID levels 5 and 6 only) • `qparity`, in which LBA range contains `qparity` (RAID level 6 only) • `data`, in which LBA range contains data
drive_lba	The LBA on the drive.
drive_start	The start of range of LBAs (strip) on the drive.
drive_end	The end of range of LBAs (strip) on the drive.
mdisk_start	The start of range of LBAs (strip) on the array MDisk.
mdisk_end	The end of range of LBAs (strip) on the array MDisk.

An invocation example

```
lsdrivelba -delim : -mdisklba 0x000 -mdisk 2
```

The resulting output:

```
drive_id:type:drive_lba:drive_start:drive_end:mdisk_start:mdisk_end
0:data:0x0000000000000000:0x0000000000000000:0x0000000000000200:0x0000000000000000:0x0000000000000200
4:parity:0x0000000000000000:0x0000000000000000:0x0000000000000200:0x0000000000000000:0x0000000000000200
```

lsdriveprogress

Use the **lsdriveprogress** command to view the progress of various drive tasks.

Syntax

```
►►—lsdriveprogress— ———————————————— ——————————————————————— ——————————————————————————►
                       └ -nohdr ┘       └ -delim — delimiter ┘

►—————————————————————————————————— ———————————————————— ——————————————————————————►◄
  └ -filtervalue — attribute=value ┘    └ -filtervalue? ┘      └drive_id┘
```

Parameters

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each

column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (**:**) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards with the SAN Volume Controller CLI:

- The wildcard character is the asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, enclose the filter entry within double quotation marks (""), as follows:

```
lsdriveprogress -filtervalue "task=*"
```

-filtervalue?
(Optional) Displays the valid filter attributes for the **-filtervalue***attribute=vaule* parameter:

- task

drive_id
(Optional) The drive for which you want to view progress.

Description

The following outputs are possible:

drive_id
Indicates the ID for the drive with the active task.

task
Indicates the type of task:

- format
- certify
- recover

progress
Indicates the percentage completion for a job.

estimated_completion_time
Indicates the estimated completion time, in the format `YYMMDDHHMMSS`, where:

- `Y` is year
- (The first) `M` is month
- `D` is day
- `H` is hour
- (The second) `M` is minute
- `S` is second

.

An invocation example

```
lsdriveprogress -delim :
```

The resulting output:

```
drive_id:task:progress:estimated_completion_time
0:format:10:091118131056
9:certify:25:991231235959
```

An invocation example

```
lsdriveprogress -delim : 9
```

The resulting output:

```
9:certify:25:991231235959
```

lsdriveupgradeprogress

Use the **lsdriveupgradeprogress** command to view the status or progress of drives with pending downloads.

Syntax

```
►►—lsdriveupgradeprogress— ┬──────────────────────┬─ ┬────────────┬──────────────────────►◄
                           └─ -delim — delimiter ─┘  └─ drive_id ─┘
```

Parameters

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum possible width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a one-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

drive_id
(Optional) Specifies the update status or progress for a single drive. If not specified, the update status for all scheduled drives is displayed.

Note: If you specify this parameter, **lsdriveupgradeprogress** displays the update status of this drive. If you do not specify this parameter, **lsdriveupgradeprogress** displays the update status of all requested drives.

Description

The **lsdriveupgradeprogress** command completes whether the original **applydrivesoftware** command was addressed to a single drive (synchronous command) or multiple drives (asynchronous command).

The following outputs are possible:

id Indicates the identity of the active drive.

status Indicates the drive status. Each of the following values has a specific meaning:

- `progressing` indicates all scheduled drives have completed; wait 270 seconds before you issue **applydrivesoftware** again.
- `completed` indicates a successful firmware download.
- `updating` indicates the update is ongoing.
- `scheduled` indicates the update is in the download list, waiting to download.
- `not_scheduled` indicates the drive is not scheduled.

Note: This means the corresponding drive was not scheduled when **`applydrivesoftware -all`** was last issued.

- `canceled` indicates the update has been canceled, either by command or by a system change that stops the **`applydrivesoftware`** command from running properly.
- `invalid` indicates the drive download status is invalid, also the initial state.
- If the field is blank, that indicates that the download is not scheduled (for example, the **`applydrivesoftware`** command has not been issued).

estimated_completion_time

Indicates the estimated completion time (`YYMMDDHHMMSS`), where:

- `Y` is year
- (The first) `M` is month
- `D` is day
- `H` is hour
- (The second) `M` is minute
- `S` is second

The value is blank if the status is either canceled or blank.

A concise invocation example with two drives canceled

```
lsdriveupgradeprogress
```

The resulting output:

```
id status    estimated_completion_time
0  completed 121112062608
5  canceled
6  canceled
```

A concise invocation example with drives scheduled or completed

```
lsdriveupgradeprogress
```

The resulting output:

```
id status    estimated_completion_time
0  completed 121112062608
5  scheduled 121112062638
6  scheduled 121112062708
```

A detailed invocation example using a drive ID

```
lsdriveupgradeprogress 17
```

The resulting output:

```
id status    estimated_completion_time
17 completed 121123134627
```

A concise invocation example

```
lsdriveupgradeprogress -delim :
```

The resulting output:

```
id:status:estimated_completion_time
0:completed:121101065019
1:scheduled:121101065049
2:scheduled:121101065119
```

A concise invocation example

```
lsdriveupgradeprogress
```

The resulting output:

```
id status    estimated_completion_time
24 completed 121212164752
25 canceled
26 canceled
```

A concise invocation example

```
lsdriveupgradeprogress
```

The resulting output:

```
id status    estimated_completion_time
0  completed 130714223913
1  completed 130714223943
2  completed 130714224013
3  completed 130714224043
4  completed 130714224113
5  completed 130714224143
6  completed 130714224213
7  completed 130714224243
8  completed 130714224313
9  completed 130714224343
10 completed 130714224413
11 completed 130714224443
```

A concise invocation example

```
lsdriveupgradeprogress -delim :
```

The resulting output:

```
id:status:estimated_completion_time
0:completed:130714223913
1:completed:130714223943
2:completed:130714224013
3:completed:130714224043
4:completed:130714224113
5:completed:130714224143
6:completed:130714224213
7:completed:130714224243
8:completed:130714224313
9:completed:130714224343
10:completed:130714224413
11:completed:130714224443
```

triggerdrivedump

Use the **triggerdrivedump** command to collect support data from a disk drive. This data can help you understand problems with the drive, and does not contain any data that applications might have written to the drive.

Syntax

```
►►— triggerdrivedump —drive_id——————————————————————————————————————►◄
```

Parameters

drive_id
(Required) The ID of the drive to dump.

Description

Use this command to collect internal log data from a drive and store the information in a file in the `/dumps/drive` directory. This directory is on one of the nodes connected to the drive. The system limits the number of drive dump files in the directory to 24 per node.

An invocation example

```
triggerdrivedump 2
```

The resulting output:

```
Dump file for drive [2] created
```

Note: The system chooses the node on which to run the statesave.

Chapter 12. Email and event notification commands

Use the email and event notification commands to enable your system to send notifications.

chemail

Use the **chemail** command to set or modify contact information for email event notifications. At least one of the parameters must be specified to modify settings.

Syntax

```
►►─ chemail ─ ┬──────────────────────────────┬ ┬──────────────────────────┬ ─────────►
              └ -reply ─ reply_email_address ┘ └ -contact ─ contact_name ┘

►─┬─────────────────────────────────────────┬ ┬──────────────────────────────────────────────┬ ──────►
  └ -primary ─ primary_telephone_number ┘     └ -alternate ─ alternate_telephone_number ┘

►─┬───────────────────────┬ ┬─────────────────────────────┬ ──────────────────────────►
  └ -location ─ location ┘  └ -contact2 ─ contact_name2 ┘

►─┬───────────────────────────────────────────┬ ┬──────────────────────────────────────────────────┬─►
  └ -primary2 ─ primary_telephone_number2 ┘     └ -alternate2 ─ alternate_telephone_number2 ┘

►─ ┬───────────────┬ ┬──────────────────────────────────┬ ┬─────────────────────┬ ────────►
   └ -nocontact2 ┘   └ -organization ─ organization ┘     └ -address ─ address ┘

►─┬───────────────┬ ┬─────────────────┬ ┬───────────┬ ┬───────────────────────┬─►◄
  └ -city ─ city ┘  └ -state ─ state ┘  └ -zip ─ zip ┘ └ -country ─ country ┘
```

Parameters

-reply *reply_email_address*
(Optional) Specifies the email address to which a reply is sent.

-contact *contact_name*
(Optional) Specifies the name of the person to receive the email.

For machine types 2071 and 2072 the maximum number of characters is 30. For other machine types the maximum number of characters is 72.

-primary *primary_telephone_number*
(Optional) Specifies the primary contact telephone number.

Note: For machine types 2071 and 2072 (in the United States and Canada), the value entered must be exactly ten decimal digits. For machines types 2071 and 2072 (in other countries) the value entered can be five to nineteen decimal digits. Otherwise, there can be up to nineteen characters.

-alternate *alternate_telephone_number*
(Optional) Specifies the alternate contact telephone number that is used when you cannot reach the primary contact on the primary phone.

-location *location*
(Optional) Specifies the physical location of the system that is reporting the error. The *location* value must not contain punctuation or any other characters that are not alphanumeric or spaces.

-contact2 *contact_name2*
(Optional) Specifies the name of the second contact person to receive the email.

For machine types 2071 and 2072 the maximum number of characters is 30. For other machine types the maximum number of characters is 72.

-primary2 *primary_telephone_number2*
(Optional) Specifies the primary contact telephone number for the second contact person.

Note: For machine types 2071 and 2072 (in the United States and Canada), the value entered must be exactly ten decimal digits. For machines types 2071 and 2072 (in other countries) the value entered can be five to nineteen decimal digits. Otherwise, there can be up to nineteen characters.

-alternate2 *alternate_telephone_number2*
(Optional) Specifies the alternate contact telephone number for the second contact person.

-nocontact2
(Optional) Removes all the contact details for the second contact person.

-organization *organization*
(Optional) Specifies the user's organization as it should appear in Call Home emails.

-address *address*
(Optional) Specifies the first line of the user's address as it should appear in Call Home email.

-city *city*
(Optional) Specifies the user's city as it should appear in Call Home email.

-state *state*
(Optional) Specifies the user's state as it should appear in Call Home email. This is a two-character value such as NY for New York.

-zip *zip*
(Optional) Specifies the user's zip code or postal code as it should appear in Call Home email.

-country *country*
(Optional) Specifies the country in which the machine resides as it should appear in Call Home email. This is a two-character value such as US for United States.

For machine types 2071 and 2072 this value cannot be US or CA if the value for **primary** or **primary2** telephone number is not blank or exactly 10 digits.

Description

This command sets or modifies contact information that is used by the email event notification facility.

Note: If you are starting the email event notification facility, the **reply**, **contact**, **primary**, and **location** parameters are required. If you are modifying contact information used by the email event notification facility, at least one of the parameters must be specified.

Remember: When considering e-mail addresses:

- Alphanumeric characters and additionally underscore (_), at symbol (@), and dot (.) characters are permitted.
- There must be exactly one @ character in the string, and the @ characters must not start or end the string.
- A plus (+) character is permitted before the @ character.

These fields do not have to be set to start the email notification system, but if the new fields are set they are included in the email event notifications.

An invocation example

```
chemail -reply ddrogba@uk.uefa.com
        -contact 'Didier Drogba'
        -primary 01962817668
        -location 'C block'
        -organization UEFA
        -address '1 Chelsea Blvd'
        -city Fulham
        -zip 0U812
        -machine_country GB
```

The resulting output:

```
No feedback
```

An invocation example

```
chemail -primary 0441234567 -location 'room 256 floor 1'
```

The resulting output:

```
No feedback
```

An invocation example

```
chemail -country US -primary 8458765309
```

The resulting output:

```
No feedback
```

chemailserver

Use the **chemailserver** command to modify the parameters of an existing email server object.

Syntax

```
►►── chemailserver ── ┬──────────────────────┬── ┬──────────────────┬── ──────────────────────────►
                      └─ -name ── server_name ┘   └─ -ip ── ip_address ┘

►─┬──────────────┬─┬─ email_server_name ─┬───────────────────────────────────────────────────►◄
  └─ -port ── port ┘ └─ email_server_id ───┘
```

Parameters

-name *server_name*
: (Optional) Specifies a unique name to assign to the email server object. The name must be a 1- through 63-character string, and cannot start with a hyphen or number. When specifying a server name, emailserver is a reserved word.

-ip *ip_address*
: (Optional) Specifies the IP address of the email server object. This must be a valid IPv4 or IPv6 address. IPv6 addresses can be zero compressed.

-port *port*
: (Optional) Specifies the port number for the email server. This must be a value of 0 - 65535. The default value is 25.

email_server_name | *email_server_id*
: (Required) Specifies the name or ID of the server object to be modified.

Description

Use this command to change the settings of an existing email server object. The email server object describes a remote Simple Mail Transfer Protocol (SMTP) email server.

You must specify either the current name or the ID of the object returned at creation time. Use the **`lsemailserver`** command to obtain this ID.

An invocation example

```
chemailserver -name newserver 0
```

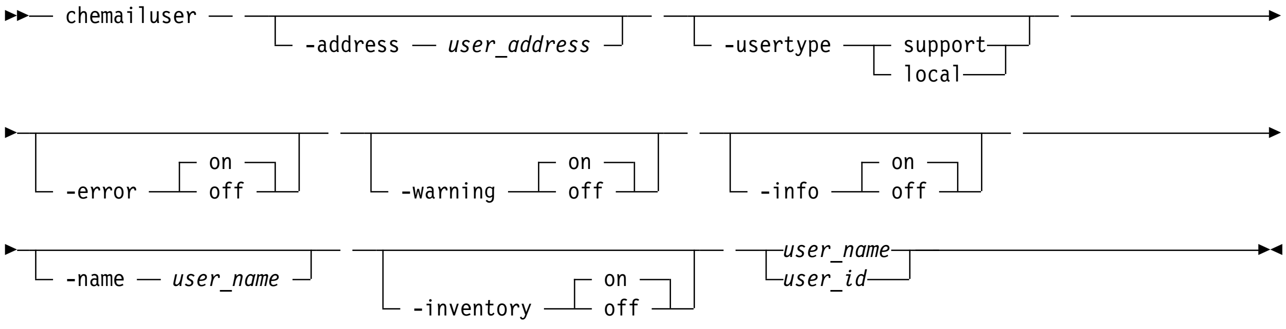
The resulting output:

```
No feedback
```

chemailuser

Use the **`chemailuser`** command to modify the settings that are defined for an email recipient.

Syntax



Parameters

`-address` *`user_address`*
(Optional) Specifies the email address of the person receiving the email or inventory notifications, or both. The *user_address* value must be unique.

`-usertype support | local`
(Optional) Specifies the type of user, either local or support, based on the following definitions:

support
Address of the support organization that provides vendor support.

local All other addresses.

`-error on | off`
(Optional) Specifies whether the recipient receives error-type event notifications. Set to `on`, error-type event notifications are sent to the email recipient. Set to `off`, error-type event notifications are not sent to the recipient.

`-warning on | off`
(Optional) Specifies whether the recipient receives warning-type event notifications. Set to `on`, warning-type event notifications are sent to the email recipient. Set to `off`, warning-type event notifications are not sent to the recipient.

-info on | off
(Optional) Specifies whether the recipient receives informational event notifications. Set to `on`, informational event notifications are sent to the email recipient. Set to `off`, informational event notifications are not sent to the recipient.

-name *user_name*
(Optional) Specifies the user name of the new email event notification recipient. The *user_name* value must be unique, must not contain spaces, and must not contain all numbers. The name **emailuser***n*, where *n* is a number, is reserved and cannot be specified as one of your user names.

-inventory on | off
(Optional) Specifies whether this recipient receives inventory email notifications.

user_name | *user_id*
(Required) Specifies the email recipient for whom you are modifying settings.

Description

This command modifies the settings that are established for an email recipient. Standard rules regarding names apply; therefore, it is not possible to change a name to **emailuser***n*, where *n* is a number.

Note: Before the **usertype** parameter can be set to `support`, the **-warning** and **-info** flags must be set to `off`.

Remember: When considering e-mail addresses:

- Alphanumeric characters and additionally underscore (_), at symbol (@), and dot (.) characters are permitted.
- There must be exactly one @ character in the string, and the @ characters must not start or end the string.
- A plus (+) character is permitted before the @ character.

An invocation example

The following example modifies email settings for email recipient `manager2008`:

```
chemailuser -usertype local manager2008
```

The resulting output:

```
No feedback
```

An invocation example

The following example modifies email settings:

```
chemailuser -address fred@gmail.com -name Fred
```

The resulting output:

```
No feedback
```

chsnmpserver

Use the **chsnmpserver** command to modify the parameters of an existing SNMP server.

Syntax

```
►►── chsnmpserver ── ──────────────────────────── ──────────────────────────────────────►
                      └─ -name ── server_name ─┘   └─ -ip ── ip_address ─┘
```

```
►─┬──────────────────────────┬─ ─┬──────────────────┬─ ─┬────────────────────┬─ ──────────►
  └─ -community ── community ─┘   └─ -error ──┬─ on ──┤   └─ -warning ──┬─ on ──┤
                                             └─ off ─┘                 └─ off ─┘

►─┬─────────────────┬─ ─┬────────────────┬─ ──┬─ snmp_server_name ─┬──────────────────►◄
  └─ -info ──┬─ on ──┤   └─ -port ── port ─┘     └─ snmp_server_id ───┘
             └─ off ─┘
```

Parameters

-name *server_name*
(Optional) Specifies a name to assign to the SNMP server. The name must be unique. When specifying a server name, *snmp* is a reserved word.

-ip *ip_address*
(Optional) Specifies an IP address to assign to the SNMP server. This must be a valid IPv4 or IPv6 address.

-community *community*
(Optional) Specifies the community name for the SNMP server.

-error on | off
(Optional) Specifies whether the server receives error notifications. Set to `on`, error notifications are sent to the SNMP server. Set to `off`, error notifications are not sent to the SNMP server.

-warning on | off
(Optional) Specifies whether the server receives warning notifications. Set to `on`, warning notifications are sent to the SNMP server. Set to `off`, warning notifications are not sent to the SNMP server.

-info on | off
(Optional) Specifies whether the server receives information notifications. Set to `on`, information notifications are sent to the SNMP server. Set to `off`, information notifications are not sent to the SNMP server.

-port *port*
(Optional) Specifies the remote port number for the SNMP server. This must be a value of 1 - 65535.

snmp_server_name | *snmp_server_id*
(Required) Specifies the name or ID of the server to be modified.

Description

Use this command to change the settings of an existing SNMP server. You must specify either the current name of the server or the ID returned at creation time. Use the **lssnmpserver** command to obtain this ID.

An invocation example

```
chsnmpserver -name newserver 0
```

The resulting output:

```
No feedback
```

lsemailserver

Use the **lsemailserver** command to display a concise list or a detailed view of email servers that are configured on the clustered system (system).

Syntax

```
►►── lsemailserver ──┬─────────┬──┬─────────────────────┬──┬───────────────────┬──────────►◄
                     └─ -nohdr ┘  └─ -delim ─ delimiter ┘  ├─ email_server_name ┤
                                                           └─ email_server_id ──┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

email_server_name | *email_server_id*
(Optional) Specifies the name or ID of an existing email server that must be listed.

Description

Use this command to display a concise list or a detailed view of email servers that are configured on the system.

A concise invocation example

```
lsemailserver -delim :
```

The concise resulting output:

```
id:name:IP_address:port
0:emailserver0:192.135.60.3:25
1:emailserver1:192.135.60.4:25
2:emailserver2:192.135.60.5:25
```

A detailed invocation example

```
lsemailserver email0
```

The detailed resulting output:

```
id 0
name emailserver0
IP_address 192.135.60.3
port 25
```

lsemailuser

Use the **lsemailuser** command to generate a report that lists the email event notification settings for all email recipients, an individual email recipient, or a specified type (local or support) of email recipient.

Syntax

```
►►─ lsemailuser ─ ┬─────────────────────┬─ ┬────────────────────┬─ ┬───────────┬──────►◄
                  └─ -type─┬─support─┬─┘   └─ -delim ─ delimiter ┘   ├─user_name─┤
                           └─local───┘                               └─user_id───┘
```

Parameters

-type support | local
(Optional) Specifies the types of email recipients you want to view, either customer-based or support-based as determined by the following definitions:

support
Address of the support organization that provides vendor support.

local All other addresses.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space that is separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, a colon separates all items of data in a concise view; the spacing of columns does not occur. In a detailed view, the data is separated from its header by a colon.

user_name | *user_id*
(Optional) Specifies the user ID or user name of the email event recipient for whom you want to see the email notification settings.

Description

When you issue this command, a report is displayed that lists the email event notification settings for all email recipients, an individual email recipient, or a specified type (local or support) of email recipient. The concise and detailed views report the same information.

A concise invocation example listing information for all email recipients by using the email event notification facility

```
lsemailuser -delim :
```

The resulting output:

```
id:name:address:user_type:error:warning:info:inventory
1:Support:callhome1@de.ibm.com:support:on:off:off:off
2:Fred:fred_house@my_company.co.uk:local:on:on:on:off
3:Log:our_log@my_company.co.uk:local:on:on:on:on
```

lssnmpserver

Use the **lssnmpserver** command to return a concise list or a detailed view of SNMP servers that are configured on the clustered system (system).

Syntax

```
►►─ lssnmpserver ─ ┬───────────┬─ ┬────────────────────┬─ ┬────────────────────┬──────►◄
                   └─ -nohdr ─┘   └─ -delim ─ delimiter ┘   ├─ snmp_server_name ─┤
                                                            └─ snmp_server_id ───┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

snmp_server_name **|** *snmp_server_id*
(Optional) Specifies the name or ID of an existing SNMP server that must be listed.

Description

Use this command to display a concise list or a detailed view of SNMP servers that are configured on the system.

A concise invocation example

```
lssnmpserver -delim :
```

The concise resulting output:

```
id:name:IP_address:error:warning:info:port:community
0:snmp0:192.135.60.4:on:on:on:78:public
1:newserver:192.136.70.7:on:off:off:250:newcommunity
```

A detailed invocation example

```
lssnmpserver snmp0
```

The detailed resulting output:

```
id 0
name snmp0
IP_address 192.135.60.4
error on
warning on
info on
port 78
community public
```

mkemailserver

Use the **mkemailserver** command to create an email server object that describes a remote Simple Mail Transfer Protocol (SMTP) email server.

Syntax

```
►►— mkemailserver — ┬─────────────────────────┬— — -ip — ip_address — ┬─────────────────┬──►◄
                    └ -name — server_name ┘                          └ -port — port ┘
```

Parameters

-name *server_name*
(Optional) Specifies a unique name to assign to the email server object. The name must be a 1- through 63-character string, and cannot start with a hyphen or number. If a name is not specified, then a system default of emailserver*n* is applied, where *n* is the object ID. When specifying a server name, emailserver is a reserved word.

-ip *ip_address*
(Required) Specifies the IP address of a remote email server. This must be a valid IPv4 or IPv6 address. IPv6 addresses can be zero compressed.

-port *port*
(Optional) Specifies the port number for the email server. This must be a value of 1 - 65535. The default value is 25.

Description

This command creates an email server object that represents the SMTP server. The SAN Volume Controller uses the email server to send event notification and inventory emails to email users. It can transmit any combination of error, warning, and informational notification types.

The SAN Volume Controller supports up to six email servers to provide redundant access to the external email network. The email servers are used in turn until the email is successfully sent from the SAN Volume Controller . The attempt is successful when the SAN Volume Controller gets a positive acknowledgement from an email server that the email has been received by the server.

An invocation example

```
mkemailserver -ip 2.2.2.2 -port 78
```

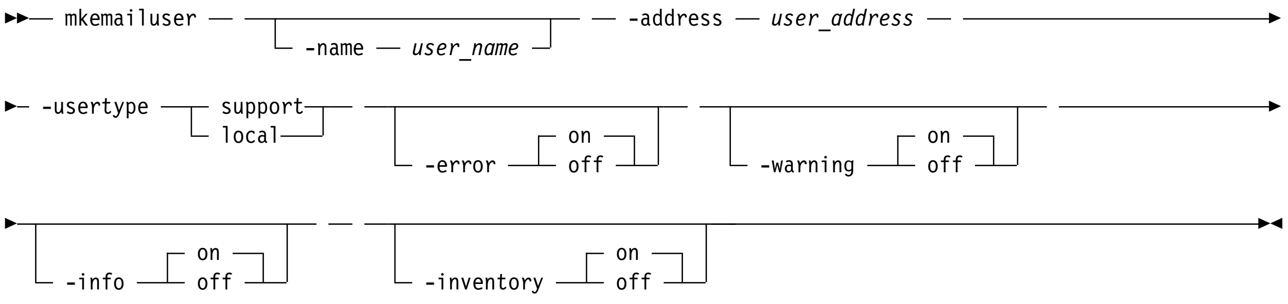
The resulting output:

```
Emailserver id [2] successfully created
```

mkemailuser

Use the **mkemailuser** command to add a recipient of email event and inventory notifications to the email event notification facility. Add up to twelve recipients (one recipient at a time).

Syntax



Parameters

-name *user_name*
(Optional) Specifies the name of the person who is the recipient of email event notifications. The *user_name* value must be unique, must not contain spaces, and must not contain only numbers. If you

do not specify a user name, the system automatically assigns a user name in the format of `emailuser`*n*, where *n* is a number beginning with 0 (`emailuser0`, `emailuser1`, and so on).

The name emailuser*n*, where *n* is a number, is reserved and cannot be used as one of your user names.

-address *user_address*
(Required) Specifies the email address of the person receiving the email event or inventory notifications, or both. The *user_address* value must be unique.

-usertype support| local
(Required) Specifies the type of user, either local or support, based on the following guidelines:

support
The recipient is your product support organization, using a default value (an auto-populated email address). This setting is used with the Call Home feature. For any other use case, contact your product support organization for direction.

local All other recipients other than your product support organization. Select the `local` usertype unless otherwise instructed by your product support organization.

-error on | off
(Optional) Specifies whether the recipient receives error-type event notifications. Set to `on`, error-type event notifications are sent to the email recipient. Set to `off`, error-type event notifications are not sent to the recipient. The default value is `on`.

-warning on | off
(Optional) Specifies whether the recipient receives warning-type event notifications. Set to `on`, warning-type event notifications are sent to the email recipient. Set to `off`, warning-type event notifications are not sent to the recipient. The default value is `on`.

-info on | off
(Optional) Specifies whether the recipient receives informational event notifications. Set to `on`, informational event notifications are sent to the email recipient. Set to `off`, informational event notifications are not sent to the recipient. The default value is `on`.

-inventory on | off
(Optional) Specifies whether this recipient receives inventory email notifications. The default value is `off`.

Description

This command adds email recipients to the email event and inventory notification facility. You can add up to twelve recipients, one recipient at a time. When an email user is added, if a user name is not specified, a default name is allocated by the system. This default name has the form of `emailuser1`, `emailuser2`, and so on. Email notification starts when you process the **startemail** command.

Note: Before you can set the **usertype** parameter to `support`, turn the **-warning** and **-info** flags `off`.

Remember: When considering e-mail addresses:

- Alphanumeric characters and additionally underscore (_), at symbol (@), and dot (.) characters are permitted.
- There must be exactly one @ character in the string, and the @ characters must not start or end the string.
- A plus (+) character is permitted before the @ character.

An invocation example

```
mkemailuser -address manager2008@ibm.com -error on -usertype local
```

The resulting output:

```
email user, id [2], successfully created
```

mksnmpserver

Use the **mksnmpserver** command to create a Simple Network Management Protocol (SNMP) server to receive notifications.

Syntax

```
►►── mksnmpserver ──┬──────────────────────────┬── -ip ── ip_address ──────────────────►
                    └── -name ── server_name ──┘

►──┬─────────────────────────────┬──┬──────────────────────┬──┬────────────────────────┬──────►
   └── -community ── community ──┘  │           ┌─ on ──┐  │  │             ┌─ on ──┐  │
                                    └─ -error ──┴─ off ─┴──┘  └─ -warning ──┴─ off ─┴──┘

►──┬─────────────────────┬──┬─────────────────┬──────────────────────────────────────►◄
   │          ┌─ on ──┐  │  └─ -port ── port ─┘
   └─ -info ──┴─ off ─┴──┘
```

Parameters

-name *server_name*
(Optional) Specifies a unique name to assign to the SNMP server. If a name is not specified, then a system default of snmp*n* is applied, where *n* is the ID of the server. When specifying a server name, snmp is a reserved word.

-ip *ip_address*
(Required) Specifies the IP address of the SNMP server. This must be a valid IPv4 or IPv6 address.

-community *community*
(Optional) Specifies the community name for the SNMP server. If you do not specify a community name, then the default name of public is used.

-error on | off
(Optional) Specifies whether the server receives error notifications. Set to on, error notifications are sent to the SNMP server. Set to off, error notifications are not sent to the SNMP server. The default value is on.

-warning on | off
(Optional) Specifies whether the server receives warning notifications. Set to on, warning notifications are sent to the SNMP server. Set to off, warning notifications are not sent to the SNMP server. The default value is on.

-info on | off
(Optional) Specifies whether the server receives information notifications. Set to on, information notifications are sent to the SNMP server. Set to off, information notifications are not sent to the SNMP server. The default value is on.

-port *port*
(Optional) Specifies the remote port number for the SNMP server. This must be a value of 1 - 65535. The default value is 162.

Description

This command creates an SNMP server to receive notifications.

SAN Volume Controller supports a maximum of 6 SNMP servers.

An invocation example

```
mksnmpserver -ip 2.2.2.2 -port 78
```

The resulting output:

```
SNMP Server id [2] successfully created
```

rmemailserver

Use the **rmemailserver** command to delete the specified email server object.

Syntax

```
►►── rmemailserver ── ──┬─ email_server_name ─┬──────────────────────────────────────────►◄
                        └─ email_server_id ───┘
```

Parameters

email_server_name | *email_server_id*
(Required) Specifies the name or ID of the email server object to be deleted.

Description

Use this command to delete an existing email server object that describes a remote Simple Mail Transfer Protocol (SMTP) email server. You must specify either the current name or the ID of the object returned at creation time. Use the **lsemailserver** command to obtain this ID.

Note: Email service stops when the last email server is removed. Use the **startemail** command to reactivate the email and inventory notification function after at least one email server has been configured.

An invocation example

```
rmemailserver email4
```

The resulting output:

```
none
```

rmemailuser

Use the **rmemailuser** command to remove a previously defined email recipient from the system.

Syntax

```
►►── rmemailuser ── ──┬user_name─┬──────────────────────────────────────────────────────►◄
                      └user_id───┘
```

Parameters

user_name | *user_id*
(Required) Specifies the user ID or user name of the email recipient to remove.

Description

This command removes an existing email recipient from the system.

An invocation example to remove email recipient `manager2008`

```
rmemailuser manager2008
```

The resulting output:

```
No feedback
```

An invocation example to remove email recipient `2`

```
rmemailuser 2
```

The resulting output:

```
No feedback
```

rmsnmpserver

Use the **rmsnmpserver** command to delete the specified Simple Network Management Protocol (SNMP)server.

Syntax

```
►►— rmsnmpserver — ─┬─ snmp_server_name ─┬──────────────────────────────────►◄
                    └─ snmp_server_id ───┘
```

Parameters

snmp_server_name | *snmp_server_id*
(Required) Specifies the name or ID of the SNMP server to be deleted.

Description

Use this command to delete an existing SNMP server. You must specify either the current name of the server or the ID returned at creation time. Use the **lssnmpserver** command to obtain this ID.

An invocation example

```
rmsnmpserver snmp4
```

The resulting output:

```
No feedback
```

sendinventoryemail

Use the **sendinventoryemail** command to send an inventory email notification to all email recipients able to receive inventory email notifications. There are no parameters for this command.

Syntax

```
►►— sendinventoryemail ——————————————————————————————————————————————►◄
```

Parameters

There are no parameters for this command.

Description

This command sends an inventory email notification to all email recipients who are enabled to receive inventory email notifications. This command fails if the **startemail** command has not been processed and at least one email recipient using the email event and inventory notification facility has not been set up to receive inventory email notifications. This command also fails if the email infrastructure has not been set up.

An invocation example

In the following example, you send an inventory email notification to all email recipients who are enabled to receive them:

```
sendinventoryemail
```

The resulting output:

```
No feedback
```

setemail (Discontinued)

Attention: The **setemail** command is discontinued. E-mail notification can be configured using the following commands: **mkemailserver**, **chemailserver**, **rmemailserver**, **chemail**, and **lsemailserver**.

startemail

Use the **startemail** command to activate the email and inventory notification function. There are no parameters for this command.

Syntax

```
►►— startemail ————————————————————————————————————————————————————————►◄
```

Parameters

There are no parameters for this command.

Description

This command enables the email event notification service. No emails are sent to users until the **startemail** command has been run and at least one user has been defined to the system.

Note: This command fails if the **chemail** command has not been used to provide adequate configuration details. The following **chemail** parameters must be specified:

- **reply**
- **contact**
- **primary**
- **location**

An invocation example to start the email error notification service

```
startemail
```

The resulting output:

```
No feedback
```

stopemail

Use the **stopemail** command to stop the email and inventory notification function. There are no parameters for this command.

Syntax

```
►►── stopemail ──────────────────────────────────────────────────────────────────►◄
```

Parameters

There are no parameters for this command.

Description

This command stops the email error notification function. No emails are sent to users until the **startemail** command is reissued.

An invocation example to stop the email and inventory notification function

```
stopemail
```

The resulting output:

```
No feedback
```

testemail

Use the **testemail** command to send an email notification to one user of the email notification function or to all users of the email notification function to ensure that the function is operating correctly.

Syntax

```
►►── testemail ──┬─user_name─┬───────────────────────────────────────────────────►◄
                 ├─user_id───┤
                 └─-all──────┘
```

Parameters

user_id | *user_name*
: (Required if you do not specify **-all**) Specifies the user ID or user name of the email recipient that you want to send a test email to. You cannot use this parameter with the **-all** parameter. The *userid_or_name* value must not contain spaces.

-all
: (Required if you do not specify *user_name* or *user_id*) Sends a test email to all email users configured to receive notification of events of any notification type. No attempt is made to send the test email to a user who does not have any notification setting set to *on*.

Description

This command sends test emails to the specified users. The email recipient expects to receive the test email within a specified service time. If the email is not received within the expected time period, the recipient must contact the administrator to ensure that the email settings for the user are correct. If there is still a problem, check your product support information.

The email recipient uses the test email to check that the Simple Mail Transfer Protocol (SMTP) name, the IP address, the SMTP port, and the user address are valid.

An invocation example that sends a test email to user ID `manager2008`

```
testemail manager2008
```

The resulting output:

```
No feedback
```

Chapter 13. Enclosure commands

Storwize V7000, Flex System V7000 Storage Node, Storwize V3500, and Storwize V3700 only: Use the enclosure commands to manage enclosures and their properties.

addcontrolenclosure

Use the **addcontrolenclosure** command to add control enclosures to the system.

Syntax

```
►►── addcontrolenclosure ── ── -iogrp ── io_grp_id_or_name ── ── -sernum ── enclosure_serial_number ───►

►─┬─────────────────────────┬──────────────────────────────────────────────────────────────────────►◄
  └─ -site ──┬─ site_name ─┬┘
             └─ site_id ───┘
```

Parameters

-iogrp ***io_grp_id_or_name***
The I/O group in which you want to put the control enclosure.

-sernum ***enclosure_serial_number***
The serial number of the control enclosure you want to add.

-site *site_name* | *site_id*
(Optional) Specifies the numeric site ID or site name of the new control enclosure. (The site ID can be set to a value of 1 or 2 only.)

Description

Use this command to add a control enclosure to the system.

Note: Transparent cloud tiering can be enabled on a system if every node on the system supports it. If a system supports transparent cloud tiering, you cannot add nodes that do not support it to the system.

Compressed or thin-deduplicated volumes can be added only to systems in which all nodes support deduplicated volumes. You can add only nodes that support deduplicated volumes to a system that contains compressed or thin deduplicated volumes. Nodes can be added only to a system that contains compressed or thin deduplicated volumes if that new node can support the amount of memory that is allocated for data deduplication in the target I/O group.

An invocation example

```
addcontrolenclosure -iogrp 0 -sernum 2361443
```

The following output is displayed:

```
Enclosure containing Node, id [x], successfully added
```

An invocation example

```
addcontrolenclosure -iogrp 1 -sernum 1234567 -site site2
```

The following output is displayed:

```
Enclosure, Node id [2], successfully added
```

chenclosure

Use the **chenclosure** command to modify enclosure properties.

Syntax

```
►►— chenclosure ——┬— -identify ——┬yes┬——┬— — —enclosure_id————————————————————————►◄
                  │              └no—┘  │
                  ├— -managed ——┬yes┬———┤
                  │             └no—┘   │
                  └— -id —enclosure_id——┘
```

Parameters

Note: Optional parameters are mutually exclusive. Exactly one of the optional parameters must be set.

-identify *yes* | *no*
(Optional) Causes the identify LED start or stop flashing.

-managed *yes*|*no*
(Optional) Changes the enclosure to a managed or unmanaged enclosure.

-id *enclosure_id*
(Optional) Changes the enclosure ID after you replace the enclosure, and enables you to control what is on the front panel.

enclosure_id
(Required) Specifies the enclosure you want to modify.

Description

Use this command to modify enclosure properties.

An invocation example to change the enclosure ID from 7 to 4

```
chenclosure -id 4 7
```

The resulting output:

```
No feedback
```

An invocation example to change enclosure 1 to unmanaged

```
chenclosure -managed no 1
```

The resulting output:

```
No feedback
```

An invocation example to make the identify LED on enclosure 1 stop flashing

```
chenclosure -identify no 1
```

The resulting output:

```
No feedback
```

chenclosurecanister

Use the **chenclosurecanister** command to modify the properties of an enclosure canister.

Syntax

```
►►── chenclosurecanister ──┬─ -excludesaport ──┬─yes─┬── -port ──┬─1─┬──┬─────────┬──┬──────────────────►
                           │                   └─no──┘           ├─2─┤  ├─ -force ─┤  │
                           │                                     ├─3─┤  └─ -reset ─┘  │
                           │                                     └─4─┘                │
                           └─ -identify ──┬─yes─┬──────────────────────────────────────┘
                                          └─no──┘

►─ -canister ──canister_id──enclosure_id────────────────────────────────────────────────────────────►◄
```

Note: Exactly one of the optional parameters must be set.

1. The **-port** and **-excludesasport** parameters must be specified together.
2. Exactly one of the optional parameters must be set.

Parameters

Note: Optional parameters are mutually exclusive.

-excludesasport *yes* | *no*
(Optional) Excludes or includes the specified SAS port. The **-port** and **-excludesasport** parameters must be specified together.

You can use the **-force** flag if there are dependent volumes.

Important: Using the **-force** flag might result in loss of access to your data.

-force
(Optional) Forces the enclosure on the canister to be excluded.

Important: Using the **-force** parameter might result in a loss of access to your data. Use it only under the direction of your product support group or representative.

-reset
(Optional) Resets the enclosure on the canister.

Important: Using the -reset parameter when the partner canister is not online can lead to a loss of access to your drives (and data). Specify `lsdependentvdisks - enclosure ID -canister ID` to determine if access to the data for one or more volumes is dependent on the canister being reset . Using the **-reset** parameter might result in a loss of access to your drives (and data). Use it only under the direction of your product support group or representative.

-identify *yes* | *no*
(Optional) Changes the state of fault light-emitting diode (LED) either to or from *slow_flashing*.

-port *1* | *2*| *3* | *4*
(Optional) Specifies the SAS port to include or exclude. The **-port** and **-excludesasport** parameters must be specified together.

Ports *3* and *4* are for Storwize V5000 only.

-canister *canister_id*
Specifies the canister to which you want to apply the change.

enclosure_id
Specifies the enclosure for which the canister is a member.

Description

This command enables you to modify the properties of an enclosure canister. You must also designate a port (using the -port parameter) when this parameter is used.

An invocation example to exclude SAS port 1 on canister 2 of enclosure 1

```
chenclosurecanister -excludesasport yes -port 1 -canister 2 1
```

The resulting output:

```
No feedback
```

An invocation example to make the fault LED flash on canister 1 of enclosure 3

```
chenclosurecanister -identify yes -canister 1 3
```

The resulting output:

```
No feedback
```

chenclosuredisplaypanel

Use the **chenclosuredisplaypanel** command to modify the properties for an enclosure display panel.

Syntax

```
►►— chenclosuredisplaypanel — ┬──────────────┬─ ┬────────────────────────────────────┬─ ───────►
                              └─ -clearswap ─┘   └─ -displaypanel — displaypanel_id ─┘

►— enclosure_id ───────────────────────────────────────────────────────────────────────────────►◄
```

Parameters

-clearswap
(Optional) Specifies that the enclosure display panel swap bit be cleared.

-displaypanel *sem_id*
(Optional) Specifies the display panel ID to change. The value must be a number.

enclosure_id
(Required) Specifies the enclosure ID for the enclosure that contains the display panel. The value must be a number 1 - 99.

Description

This command modifies the properties for an enclosure display panel.

An invocation example

```
chenclosuredisplaypanel -clearswap -displaypanel 1 3
```

The resulting output:

```
No feedback
```

chenclosuresem

Use the **chenclosuresem** command to modify the properties of an enclosure SEM.

Syntax

```
►►— chenclosuresem — ┬───────────────┬─ ┬──────────────────────────────┬─ — enclosure_id ———————►◄
                     └─ -clearswap ─┘  └─ -enclosuresemid — sem_id ─┘
```

Parameters

-clearswap
(Optional) Specifies that the enclosure SEM swap bit be cleared.

-enclosuresemid *sem_id*
(Optional) Specifies the enclosure SEM ID. The value must be a number 1 - 2.

enclosure_id
(Required) Specifies the enclosure ID for the enclosure that contains the SEM. The value must be a number 1 - 99.

Description

This command modifies the properties of an enclosure SEM.

An invocation example

```
chenclosuresem -clearswap -enclosuresemid 1 8
```

The resulting output:

```
No feedback
```

chenclosureslot

Use the **chenclosureslot** command to modify the properties of an enclosure slot.

Syntax

```
►►— chenclosureslot ─┬─ -identify ─┬─ yes ─┬──────────────────────────────────────┬──────────►
                     │             └─ no ──┘                                      │
                     └─ -exclude ─┬─ yes ─┬─┬────────────────────┬─┬────────────┬─┘
                                  └─ no ──┘ └─ -port — port_id ─┘ └─ -force ─┘

►— -slot— slot_id — enclosure_id ——————————————————————————————————————————————————►◄
```

Note:

1. Optional parameters are mutually exclusive.
2. You can only specify **-port** or **-force** when you also specify **-exclude**.
3. Exactly one of the optional parameters must be set.
4. Using **-force** has an effect on the operation of **-exclude** *yes*.

Parameters

-identify *yes* | *no*
(Optional) Change the state of fault light-emitting diode (LED) to or from *slow_flashing*.

-exclude *yes* | *no*
(Optional) Ensures that an enclosure slot port is excluded. The following list gives details of the options you can use with this parameter:

- **-exclude** *yes* **-port** *port_id***-slot** *slot_id enclosureid*: The port you specify with *port_id* is excluded. If the current state of the port is `excluded_by_enclosure`, `excluded_by_drive`, or `excluded_by_cluster`, this command will appear to have no effect. However, if the current state of the port is `online`, then that state will change to `excluded_by_cluster`. The port will remain excluded until you rerun this command with *no* selected.

  Attention: This command checks for dependent volumes. If issuing this command would result in losing access to data, then the command fails and an error message displayd. You can use the **-force** flag to ignore these errors, but this could result in loss of access to data.
- **-exclude** *no* **-port** *port_id***-slot** *slot_id enclosureid*: The port is put into `online` state, provided there are no other reasons to exclude the port. If you issue this command when the port is `online`, then it has no effect. However, if you issue this command when the port is `excluded`, then the port state will do one of the following:
  - Change to `online` status immediately.
  - Change to `online` status after all other reasons for the port to be excluded have been removed.
- **-exclude** *yes* | *no* **-slot***slot_id enclosureid*: If you issue this command without defining a port, the command simultaneously acts on both ports.

-port *1* | *2*
(Optional) Specifies the port on the canister to be excluded. If it is not specified, **-exclude** acts on both ports.

-force
(Optional) Forces the port on the canister to be excluded.

Important: Using the **-force** parameter might result in a loss of access. Use it only under the direction of your product support information.

-slot *slot_id*
(Required) Specifies the slot ID. The value must be a number from 1 - 92.

The slots are numbered 1 (leftmost) to 24 (rightmost) when viewed from in front of the enclosure, in 24-slot enclosures. In 12-slot enclosures, the slots are arranged in numerical order in three rows with four slots. For example, the:

- First row contains slots 1, 2, 3, and 4 (in that order)
- Second row contains slots 5, 6, 7, and 8 (in that order)
- Third row contains slots 9, 10, 11, and 12 (in that order)

enclosure_id
(Required) The enclosure that the slot is a member of.

Description

These commands enable you to modify the properties of an enclosure slot.

An invocation example to turn on the identify LED on slot 7 in enclosure 1

```
chenclosureslot -identify yes -slot 7 1
```

The resulting output:

```
No feedback
```

An invocation example to force the exclusion of port 1 of slot 7 in enclosure 3

```
chenclosureslot -exclude yes -port 1 -force -slot 7 3
```

The resulting output:

```
No feedback
```

(satask) chenclosurevpd (Deprecated)

The **chenclosurevpd** command is deprecated. Use the **chvpd** command instead.

lsenclosure

Use the **lsenclosure** command to view a summary of the enclosures.

Syntax

```
►►── lsenclosure ── ┬─────────┬─ ┬─────────────────────────────────┬─ ┬───────────────┬──►
                    └ -nohdr ─┘  └ -filtervalue ── attribute_value ┘  └ -filtervalue? ┘

►─ ┬─────────────────────────┬─ ┬──────────────┬─ ──────────────────────────────────────►◄
   └ -delim ── delimiter ────┘  └─enclosure_id─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If no data is available to be displayed, headings are not displayed.

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards:

- The wildcard character is the asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, enclose the filter entry within double quotation marks (""): `lsenclosure -filtervalue id="1*"`

-filtervalue?
(Optional) Displays the valid filter attributes for the **-filtervalue** parameter:

- drive_slots
- id
- IO_group_id
- IO_group_name
- managed
- online_canisters
- online_PSUs
- product_MTM
- serial_number
- status
- total_canisters
- total_PSUs
- type

-delim *delimiter*

(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

enclosure_id

Detailed information for the enclosure that you specify.

Description

This command displays a summary of the enclosures (including status information for canisters and power and cooling units, and other enclosure attributes). Table 62 shows the possible outputs:

Table 62. lsenclosure output

Attribute	Description
id	Indicates the ID of the enclosure.
status	Indicates whether an enclosure is visible to the SAS network: • `online` if a managed or unmanaged enclosure is visible. • `offline` if a managed enclosure is not visible, and other fields hold their last known values. • `degraded` if an enclosure is visible, but not down both strands.
type	Indicates the type of enclosure: • `control` • `expansion`
managed	Indicates whether the enclosure is managed: • `yes` • `no`
IO_group_id	Indicates the I/O group the enclosure belongs to; blank if canisters are connected to two different I/O groups.
IO_group_name	Indicates the I/O group the enclosure belongs to; blank if canisters are connected to two different I/O groups.
fault_LED	Indicates the status of the fault light-emitting diode (LED) on the enclosure: • `on` if a service action is required immediately on the enclosure or a component within the enclosure (including a canister, power unit, or non-spared drive). • `slow_flashing` if battery power not sufficient to run I/O. • `off` if faults exist on the enclosure or its components.
identify_LED	Indicates the state of the identify LED: • `off` if the enclosure is not identified. • `slow_flashing` if the enclosure is being identified.
error_sequence_number	Indicates the error log number of the highest priority error for this object. This attribute is typically blank; however, if a problem exists (for example, the status has degraded), then it contains the sequence number of that error.
product_MTM	Indicates the product machine type and model.
serial_number	Indicates the serial number of the enclosure. This serial number is the product serial number, which indicates the enclosure and its contents. The enclosure has its own serial number, which is embedded in the FRU_identity 11S data.
FRU_part_number	Indicates the FRU part number of the enclosure.

Table 62. lsenclosure output (continued)

Attribute	Description
FRU_identity	Indicates the 11S serial number that combines the manufacturing part number and the serial number.
total_canisters	Indicates the maximum number of canisters for this enclosure type.
online_canisters	Indicates the number of canisters that are contained in this enclosure that are online.
total_PSUs	Indicates the number of power and cooling units in this enclosure.
online_PSUs	Indicates the number of power-supply units (PSUs) contained in this enclosure that are online.
drive_slots	Indicates the number of drive slots in the enclosure.
firmware_level_1	Indicates the version of the microcode image (midplane firmware version) installed on the midplane.
firmware_level_2	Indicates the version of the midplane metadata (midplane vital product data, or VPD, version) installed on the midplane.
machine_part_number	Blank.
machine_signature	Indicates a machine signature unique to the control enclosure and representing the serial number and machine part number. The format is a hyphenated string of 19 hexadecimal characters. **Remember:** Expansion enclosures do not have a machine signature.
interface_speed	Indicates the SAS interface speed of the enclosure (in gigabits per second, or Gbps). The values are: • 6 Gbps • 12 Gbps • Blank for an unknown or unsupported enclosure
total_sems	Indicates the total number of secondary expander modules (SEMs) that are in the system. The value must be a number 0 - 2.
online_sems	Indicates the total number of SEMs in the system that are online. The value must be a number 0 - 2.

A detailed invocation example

```
lsenclosure 1
```

The following detailed output is displayed:

```
id 1
status online
type control
managed no
IO_group_id 0
IO_group_name io_grp0
fault_LED off
identify_LED off
error_sequence_number
product_MTM 2072-02A
serial_number 64G005S
FRU_part_number 85Y5896
FRU_identity 11S85Y5962YHU9994G005S
total_canisters 2
online_canisters 2
total_PSUs 2
online_PSUs 2
drive_slots 12
firmware_level_1 10
firmware_level_2 F6C07926
machine_part_number 2072L2C
```

```
machine_signature 0123-4567-89AB-CDEF
ambient_temperature 30
total_fan_modules:2
online_fan_modules:2
interface_speed:6Gb
total_sems 2
online_sems 1
```

lsenclosurebattery

Use the **lsenclosurebattery** command to display information about the batteries. The batteries are located in the node canisters.

Syntax

```
►►── ── lsenclosurebattery ── ─┬─────────┬─ ─┬──────────────────────────────────┬─ ──────────►
                               └─ -nohdr ┘   └─ -filtervalue ── attribute_value ┘

►─┬─────────────────┬─ ─┬──────────────────────┬─ ──────────────────────────────────────────►
  └─ -filtervalue? ─┘   └─ -delim ── delimiter ┘

►─┬──────────────────────────────────────────────┬─ ────────────────────────────────────►◄
  └─┬───────────────────────────┬─ enclosure_id ─┘
    └─ -battery ── battery_id ──┘
```

Parameters

-nohdr
: (Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

  Note: If there is no data to be displayed, headings are not displayed.

-filtervalue *attribute=value*
: (Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are returned. If a capacity is specified, the units must also be included.

  Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards when you use the CLI:

  - The wildcard character is an asterisk (*).
  - The command can contain a maximum of one wildcard.
  - When you use a wildcard, you must enclose the filter entry within double quotation marks (""), as follows:

    ```
    lsenclosurebattery -filtervalue "battery_id=1"
    ```

-filtervalue?
: (Optional) Displays a list of valid filter attributes for the **-filtervalue** *attribute=value* parameter:

  - battery_id
  - charging_status
  - enclosure_id
  - end_of_life_warning
  - percent_charged
  - recondition_needed
  - status

-delim *delimiter*

(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. A detailed view provides each item of data in its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-battery *battery_id*

(Optional) Provides a detailed view of the specified battery. Valid only when an enclosure is specified.

enclosure_id

(Optional) Lists the batteries for the specified enclosure.

Description

This command displays information about the batteries, which are located in the node canisters. The concise view displays a line for each battery slot in every enclosure whether a battery exists for that slot or not. Batteries are not shown for expansion enclosures.

Table 63 shows possible outputs.

Table 63. lsenclosurebattery outputs

Attribute	**Description**
enclosure_id	Identifies the identity of the enclosure that contains the battery.
battery_id	Identifies the battery in the enclosure.
status	Identifies the status of the battery: • `online` indicates that the battery is present and working as usual. • `degraded` indicates that the battery is present but not working as usual. • `offline` indicates that the battery cannot be detected.
recondition_needed	Identifies that the battery needs to be reconditioned or it must start reconditioning soon. **Remember:** If this message is permanent, there might be an error that prevents the recondition from starting.
percent_charged	Indicates the charge of battery (in a percentage).
end_of_life_warning	Identifies the battery's end of life (with a warning noise). The values are `yes` and `no`. **Important:** Replace the battery.
FRU_part_number	Identifies the FRU part number of the battery.
FRU_identity	Identifies the 11S number, combining the manufacturing part number and the serial number.
firmware_level	The version of the microcode image that is installed on the battery.
error_sequence_number	Indicates the error log (or event log) number of the highest priority error for this object. This output field is typically blank. However, if there is a problem (for example, the status is `degraded`), then it contains the sequence number of that error event.
remaining_charge_capacity_mAh	Identifies the battery's remaining charge capacity in milliampere-hour (mAh).
full_charge_capacity_mAh	Identifies the fully charged capacity of the battery in mAh (this value diminishes as the battery ages).
compatibility_level	Identifies the battery driver software must support this level to operate with this battery - this value comes from the battery vital product data (VPD).

Table 63. lsenclosurebattery outputs (continued)

Attribute	Description
last_recondition_timestamp	Identifies a system timestamp, in the format `YYMMDDHHMMSS`, for when the last successful calibration of the gas gauge occurred, where: • `Y` is year • (The first) `M` is month • `D` is day • `H` is hour • (The second) `M` is minute • `S` is second
powered_on_hours	Identifies the number of hours the battery is in a powered node (not necessarily the same node).
cycle_count	Identifies the number of charge or discharge cycles that are performed on the battery.

A concise invocation example

```
lsenclosurebattery 1
```

The resulting output:

```
enclosure_id battery_id status charging_status recondition_needed percent_charged end_of_life_warning
1            1          online idle            no                 100             no
1            2          online idle            no                 100             no
```

A detailed invocation example

```
lsenclosurebattery -battery 1 1
```

The resulting output:

```
enclosure_id 1
battery_id 1
status online
charging_status idle
recondition_needed no
percent_charged 100
end_of_life_warning no
FRU_part_number 31P1807
FRU_identity 11S00AR085YM30BG42R04P
firmware_level 105:1
error_sequence_number
remaining_charge_capacity_mAh 3477
full_charge_capacity_mAh 3795
compatibility_level 1
last_recondition_timestamp 140528045617
powered_on_hours 1162
cycle_count 10
```

lscontrolenclosurecandidate (Storwize family products only)

Use the **`lscontrolenclosurecandidate`** command to display a list of all control enclosures you can add to the current system.

Syntax

```
►►— lscontrolenclosurecandidate — ─┬─────────┬─ ─┬──────────────────────┬─ ──────────────────►◄
                                   └─ -nohdr ─┘   └─ -delim — delimiter ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

Description

Table 64 provides the possible values that are applicable to the attributes that are displayed as data in the output views.

Table 64. lscontrolenclosurecandidate attribute values

Attribute	Value
serial_number	Indicates the serial number for the enclosure.
product_MTM	Indicates the MTM for the enclosure.
machine_signature	Indicates the machine signature for the enclosure.

A concise invocation example

```
lscontrolenclosurecandidate
```

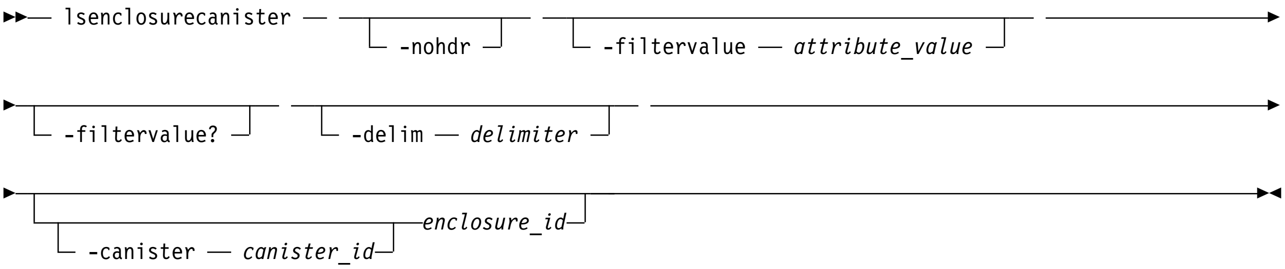
The concise resulting output:

```
serial_number product_MTM machine_signature
G00F7GY       2076-624 5746-9812-B5CF-FEF9
```

lsenclosurecanister

Use the **lsenclosurecanister** command to view a detailed status for each canister in an enclosure.

Syntax



Parameters

-nohdr

(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-filtervalue *attribute=value*

(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are returned. If a capacity is specified, the units must also be included.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards when you use the CLI:

- The wildcard character is an asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, you must enclose the filter entry within double quotation marks (""), as follows:

```
lsenclosurecanister -filtervalue "node_name=node*"
```

-filtervalue?

(Optional) Displays a list of valid filter attributes for the **-filtervalue** *attribute=value* parameter:

- enclosure_id
- canister_id
- node_id
- node_name
- status
- type

-delim *delimiter*

(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-canister *canister_id*

Valid only when the enclosure_id is specified. Provides a detailed view of the canister for the specified enclosure.

enclosure_id

Lists the canisters for the specified enclosure.

Description

This command displays a detailed status for each canister in an enclosure. Table 65 shows the possible outputs:

Table 65. lsenclosurecanister output

Attribute	**Description**
enclosure_id	Indicates the enclosure that contains the canister.
canister_id	Indicates which of the canisters in the enclosure it is.

Table 65. lsenclosurecanister output (continued)

Attribute	Description
status	Indicates the status of the canister. The values are: • online indicates that the canister is present and working normally. • degraded indicates that the canister is present but not working normally • offline indicates that the canister could not be detected.
type	Indicates the type of canister. The values are node or expansion.
node_id	Indicates the node that corresponds to this canister; blank if the canister is not a node, or if the node is offline or not part of the clustered system.
node_name	Indicates the node that corresponds to this canister; blank if the canister is not a node, or if the node is offline or not part of the clustered system.
temperature	Indicates the temperature of the canister in degrees centigrade.
identify_LED	Indicates the status of the identify_LED. The values are on, off, or slow-flashing.
fault_LED	Indicates the status of the fault_LED. The values are on, off, or slow-flashing.
SES_status	Indicates the Small Computer System Interface (SCSI) status of the connection between the device and the canister. The values are online and offline.
FRU_part_number	Indicates the field-replaceable unit (FRU) part number of the canister.
FRU_identity	Indicates the 11S number that combines the manufacturing part number and the serial number.
WWNN	Indicates the Fibre Channel (FC) worldwide node name (WWNN) of the canister (node canisters only).
temperature	Indicates thehe temperature of the canister (in degrees Celsius). If the temperature goes below 0, s0 is displayed. The value must be in the range 0 - 245. **Remember:** The temperature value is not an ambient temperature value. It is an internal temperature sensor value.
fault_LED	(0 to 245) state of the combined fault and identify light-emitting diodes (LEDs): • off indicates that it is not a fault • slow_flashing indicates that it is identify mode **Note:** When the LED is in identify mode, it conceals whether there is a fault present, because it always flashes. When you remove it from identity mode, the LED becomes on or off. • on indicates that it is a fault
error_sequence_number	Indicates the error log number of the highest priority error for this object. This is typically blank; however, if there is a problem (for example, the status is degraded), then it contains the sequence number of that error.
SAS_port_1_status	Indicates whether there is damage to the cable between SAS ports: • online • offline • excluded (meaning logged in but cannot communicate with the canister) • degraded (meaning the SAS cable is not fully functional) • Blank (which can appear on control canisters; see **lsportsas**)
SAS_port_2_status	Indicates whether there is damage to the cable between SAS ports: • online • offline • excluded (meaning logged in but cannot communicate with the canister) • degraded (meaning the SAS cable is not fully functional) • Blank (which can appear on control canisters; see **lsportsas**)

Table 65. lsenclosurecanister output (continued)

Attribute	Description
firmware_level	Indicates the firmware level of the Small Computer System Interface (SCSI) Enclosure Services (SES) code, or canister firmware version, running on the canister.
firmware_level_2	Indicates the version of the first other microcode image (canister bootloader version) installed on the canister.
firmware_level_3	Indicates the version of the second other microcode (canister complex programmable logic device, or CPLD, version) image installed on the canister.
firmware_level_4	Indicates the version of the third other microcode image (canister flash configuration version) installed on the canister.
firmware_level_5	Indicates the version of the canister metadata (canister VPD version) installed on the canister.

A concise invocation example

```
lsenclosurecanister -delim :
```

The resulting output:

```
enclosure_id:canister_id:status:type:node_id:node_name
1:1:degraded:expansion:1:node1
```

A detailed invocation example

```
lsenclosurecanister -canister 1 1
```

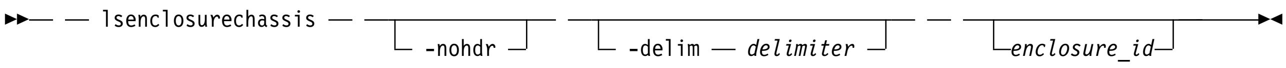
The detailed resulting output:

```
enclosure_id 1
canister_id 1
status online
type node
node_id 1
node_name node1
FRU_part_number AAAAAAA
FRU_identity 11S1234567Y12345678901
WWNN 5005076801005F94
firmware_level XXXXXXXXXX
temperature 23
fault_LED flashing
SES_status online
error_sequence_number
SAS_port_1_status online
SAS_port_2_status online

firmware_level_2 0501
firmware_level_3 14
firmware_level_4 B69F66FF
firmware_level_5 5C2A6A44
```

lsenclosurechassis

Use the **lsenclosurechassis** command to provide a description of the chassis-specific enclosure properties, including its location within the chassis.

Syntax



```
►►— — lsenclosurechassis — ─────────── ──────────────────── — ───────────────►◄
                            └ -nohdr ┘  └ -delim — delimiter ┘    └enclosure_id┘
```

Parameters

-nohdr

(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*

(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim : on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

enclosure_id

(Optional) Indicates the unique enclosure identifier (a number in the range 1 - 99).

Description

This command has both a detailed and concise view. The *enclosure_id* keyword is required for the detailed view.

The following table displays information about chassis-specific enclosure properties and shows possible outputs.

Table 66. lsenclosurechassis outputs

Attribute	Description
enclosure_id	Specifies the enclosure identifier. It is a numeric character between the numbers 1 and 99.
chassis_name	Specifies the chassis name. It can be set from the CMM, and is blank or an alphanumeric string that contains up to 128 characters.
canister_1_bay	Specifies the first canister bay's enclosure position within the chassis. It is a numeric character between the numbers 0 and 254.
canister_2_bay	Specifies the second canister bay's enclosure position within the chassis. It is a numeric character between the numbers 0 and 254.
numbering scheme	Specifies the chassis numbering scheme set from the CMM. It can be a numeric character between the numbers 0 and 255.
pos_in_rack	Specifies the chassis position within the rack set from the CMM. It must be an alphanumeric 2-character string.
rack_location	Specifies the location of the rack that contains the chassis set from the CMM. It can be blank or an alphanumeric string that contains up to 128 characters.
rack_room	Specifies the room that contains the rack set from the CMM. It can be blank or an alphanumeric string that contains up to 128 characters.
chassis_mtm	Specifies the chassis machine type or model. The type or model is an alphanumeric string that contains up to 22 characters.
chassis_sn	Specifies the chassis serial number. The serial number is an alphanumeric string that contains up to 22 characters.
chassis_uuid	Specifies the chassis unique user identifier. The identifier is an alphanumeric string that contains up to 128 characters.

Table 66. lsenclosurechassis outputs (continued)

Attribute	Description
chassis_rack	Specifies the identifier for the rack that contains the chassis. The identifier is blank or an alphanumeric string that contains up to 128 characters.

An invocation example

```
lsenclosurechassis 1
```

The resulting output:

```
enclosure_id 1
chassis_name 25631
 canister_1_bay 0
canister_2_bay 0
numbering_scheme 0
pos_in_rack 1
rack_location In the corner
rack_room D-East
chassis_mtm 2078-219
chassis_sn 64H123R
chassis_uuid 987654321
chassis_rack Rack47
```

lsenclosuredisplaypanel

Use the **lsenclosuredisplaypanel** command to display information about the display panel in an enclosure.

Syntax

```
►►── lsenclosuredisplaypanel ── -displaypanel ── displaypanel_id ── ┬──────────┬ ──────────────►
                                                                     └ -nohdr ─┘

►─┬──────────────────────┬─ ┬────────────────┬──────────────────────────────────────────────►◄
  └ -delim ── delimiter ─┘  └ enclosure_id ─┘
```

Parameters

-displaypanel *displaypanel_id*
: (Required) Specifies the display panel ID for the display panel that is being displayed. The value must be a number.

-nohdr
: (Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

  Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
: (Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

enclosure_id
(Required) Specifies the enclosure ID for the enclosure data that is being displayed. The value must be a number 1 - 99.

Description

This command displays information about the display panel in an enclosure.

This table provides the attribute values that can be displayed as output view data.

Table 67. lsenclosuredisplaypanel output

Attribute	Description
enclosure_id	Indicates the enclosure ID for the enclosure that contains the display panel. The value must be a number 1 - 99.
displaypanel_id	Indicates the ID of the display panel that is in the enclosure. The value must be a number.
status	Indicates the display panel status for the display panel that is in the enclosure. The values are: • `enum` • `online` • `degraded` • `offline`
error_sequence_number	Indicates the event log sequence number for the current event that is logged against the secondary expander module (SEM). The value is blank if there is no event to log.
FRU_part_number	Indicates the FRU part number of the display panel. The value must be a 7-character numeric string.
FRU_identity	Indicates the FRU identity of the display panel. The value must be a 22-character alphanumeric string.

A concise invocation example

```
lsenclosuredisplaypanel
```

The resulting output:

```
enclosure_id display_panel_id status
1            1                online
2            1                online
3            1                online
```

A detailed invocation example

```
lsenclosuredisplaypanel -displaypanel 1 3
```

The resulting output:

```
enclosure_id 3
displaypanel_id 1
status online
error_sequence_number
FRU_Part_Number *******
FRU_Identity **********************
```

lsenclosurefanmodule

Use the **lsenclosurefanmodule** command to report the status of each fan module and the contained fans in an enclosure.

Syntax

```
►►──lsenclosurefanmodule── ─┬─────────┬─ ─┬──────────────────────┬─ ──────────────────────►
                            └─ -nohdr ─┘   └─ -delim ── delimiter ─┘

►─┬──────────────────────────────────────────┬──────────────────────────────────────────►◄
  └─┬─────────────────────────────┬─enclosure_id─┘
    └─-fanmodule──fanmodule_id─┘
```

Parameters

-fanmodule *fanmodule_id*
(Optional) Specifies the ID of the fan module for which data is displayed. The possible values are 1 or 2, and any other value returns no output.

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

enclosure_id
(Optional) Specifies the ID of the enclosure for which data is displayed.

Description

The command reports status for fan modules and the contained fans in an enclosure.

Table 68 provides the possible values that are applicable to the attributes that are displayed as data in the output views.

Table 68. lsenclosurefanmodule attribute values

Attribute	**Value**
enclosure_id	Indicates the enclosure ID of the enclosure that contains the fan module.
fan_module_id	Indicates the fan module ID of the fan module that is in the enclosure. The possible values are 1 or 2.
status	Indicates the combined status of the fan module and any contained fans. The values are: • `online` • `offline` • `degraded`
error_sequence_number	Indicates the event log sequence number of the current event that is logged against the fan module. It is blank if there is no current event.
FRU_part_number	Indicates the part number of the fan module.
FRU_identity	Indicates the FRU identity of the fan module.

Table 68. lsenclosurefanmodule attribute values (continued)

Attribute	Value
fault_LED	Indicates the status of the fault light-emitting diode (LED) on the fan module: • `on`, which indicates that the LED is on • `off`, which indicates that the LED is off • `unknown`, which indicates that the LED status is unknown

An invocation example

```
lsenclosurefanmodule
```

The resulting output:

```
enclosure_id fan_module_id status
1            1             online
1            2             online
2            1             online
2            2             online
```

An invocation example

```
lsenclosurefanmodule 2
```

The resulting output:

```
enclosure_id fan_module_id status
2            1             online
2            2             online
```

An invocation example

```
lsenclosurefanmodule -fanmodule 1 1
```

The resulting output:

```
enclosure_id 1
fan_module_id 1
status online
error_sequence_number
FRU_part_number 31P1847
FRU_identity 11S31P1846YM10BG3B101N
fault_LED off
```

lsenclosurepsu

Use the **lsenclosurepsu** command to view information about each power-supply unit (PSU) in the enclosure.

Syntax

```
►►── lsenclosurepsu ── ──────────── ───────────────────────────────── ─────────────────────────►
                      └─ -nohdr ─┘  └─ -filtervalue ── attribute_value ─┘

►─┬─────────────────┬─ ─┬──────────────────────┬─ ─┬──────────────────────────────────┬──────►◄
  └─ -filtervalue? ─┘   └─ -delim ── delimiter ─┘   └─┬──────────────────┬─ enclosure_id ─┘
                                                      └─ -psu ── psu_id ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards:

- The wildcard character is the asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, enclose the filter entry within double quotation marks (""):

```
lsenclosurepsu -filtervalue "psu_id=1"
```

-filtervalue?
(Optional) Displays the valid filter attributes for the **-filtervalue** parameter:

- enclosure_id
- psu_id
- status

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-psu *psu_id*
(Optional) Valid only when the **enclosure_id** is specified. Provides a detailed view of the PSU for the specified enclosure.

enclosure_id
(Optional) Lists the PSUs for the specified enclosure.

Description

This command displays information about each power-supply unit (PSU) in the enclosure. Table 69 shows the possible outputs:

Table 69. lsenclosurepsu output

Attribute	Description
enclosure_id	Indicates the ID of the enclosure that contains the PSU.
psu_id	Indicates the ID of the PSU in the enclosure.
status	Indicates the status of the power and cooling unit in the enclosure: • `online` indicates that a PSU is present and working normally. • `offline` indicates that a PSU cannot be detected. • `degraded` indicates that a PSU is present but not working normally.

Table 69. lsenclosurepsu output (continued)

Attribute	Description
input_failed	• `on` indicates that no usable input power is detected from the power distribution unit. • `off` indicates that the input power is OK.
output_failed	• `on` indicates that no usable output power is detected from the distribution unit. • `off` indicates that the output power is OK.
input_power	• `ac` indicates that the power supply requires AC power input. • `dc` indicates that the power supply requires DC power input. • `unknown` indicates that the power supply is not known or cannot be determined.
fan_failed	• `on` indicates that if the AC, DC, and fan LEDs are all on, there is a PSU fault. If only the fan LED is on, then there is a fan failure. • `off` indicates that the fans in this PSU are OK.
redundant	Indicates (`yes` or `no`) whether you can remove the power supply: • If the PSU is on an expansion enclosure, the other PSU must be online. • If the PSU is on a control enclosure, the other PSU must be online and the battery on that PSU must contain enough charge to allow the canisters to dump state and cache data before you shut down.
error_sequence_number	Indicates the error log (or event log) number of the highest priority error for this object. This value is typically blank. However, if there is a problem (for example, the status is degraded), then it contains the sequence number of that error event.
FRU_part_number	Indicates the FRU part number of the PSU.
FRU_identity	Indicates the 11S number, combining the manufacturing part number and the serial number.
firmware_level_1	Indicates the version of the microcode image (power supply firmware version) installed on the power supply.
firmware_level_2	Indicates the version of the power supply metadata (power supply vital product data, or VPD, version) installed on the power supply. **Note:** This field might not be applicable for some systems and is blank for all PSU types.
firmware_level_3	Indicates the version of the secondary microcode image that is installed on the enclosure's High Efficiency (HE) power supply unit (PSU). **Note:** This field might not be applicable for some systems and is blank for all PSU types.

An invocation example

```
lsenclosurepsu -delim :
```

The resulting output:

```
enclosure_id:PSU_id:status:input_power
1:1:online:ac
1:2:online:ac
```

A detailed invocation example

```
lsenclosurepsu -psu 1 1
```

The detailed resulting output:

```
enclosure_id 1
PSU_id 1
status online
input_failed off
```

```
output_failed on
fan_failed off
redundant yes
error_sequence_number
FRU_part_number 85Y5847
FRU_identity 11S85Y5847YG50CG07W0LJ
firmware_level_1 0314
firmware_level_2 AF9293E5
firmware_level_3
input_power ac
```

lsenclosuresem

Use the **lsenclosuresem** command to display the status (or any pertinent data) about secondary expander modules (SEM) in a 5U92 system.

Syntax

```
►►── lsenclosuresem ── ┬──────────────────┬ ┬──────────┬ ┬──────────────────────┬ ──────────►
                       └─ -sem ── sem_id ─┘ └─ -nohdr ─┘ └─ -delim ── delimiter ─┘

►─ enclosure_id ─────────────────────────────────────────────────────────────────────────►◄
```

Parameters

-sem *sem_id*
(Optional) Specifies the SEM ID for the SEM data that is being displayed.

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim : on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

enclosure_id
(Required) Specifies the enclosure ID for the enclosure data that is being displayed. The value must be an integer 1 - 99.

Description

This command displays the status (or any pertinent data) about SEM in a 5U92 system.

A 5U92 system is a 5U enclosure that can contain up to 92 3.5-inch drives (but must be used in an expansion enclosure).

This table provides the attribute values that can be displayed as output view data.

Table 70. lsenclosuresem output

Attribute	**Description**
id	Indicates the enclosure ID of the enclosure that contains the disk drawer. The value must be a number 1 - 99.
sem_id	Indicates the SEM ID for the SEM that is in the enclosure. The value must be a number, 1 or 2.
status	Indicates the SEM status for the SEM that is in the enclosure. The values are: • `online, which indicates that the SEM is online` • `degraded, which indicates that the SEM is degraded` • `offline, which indicates that the SEM is offline`
expander_1_status	Indicates the status of the first or lowest order expander index. The values are: • `online, which indicates that the SEM is online` • `degraded, which indicates that the SEM is degraded` • `offline, which indicates that the SEM is offline`
expander_2_status	Indicates the status of the second expander index (or *e+1*, where *e* is the lowest-order or first expander index). The values are: • `online, which indicates that the SEM is online` • `degraded, which indicates that the SEM is degraded` • `offline, which indicates that the SEM is offline`
error_sequence_number	Indicates the event log sequence number of the current event that is logged against the SEM. The value is blank if there is no error to log.
FRU_part_number	Indicates the field-replaceable unit (FRU) part number for the SEM. The value must be a 7-character numeric string.
FRU_identity	Indicates the FRU ID for the SEM. The value must be a 22-character alphanumeric string.
firmware_level_1	Indicates the SCSI Enclosure Services (SES) firmware level of the lowest order expander index. The value must be a 22-character alphanumeric string.
firmware_level_2	Indicates the bootloader firmware level of the lowest order expander index. The value must be a 22-character alphanumeric string.
firmware_level_3	Indicates the SES firmware level of the second lowest order expander index. (or *s+1*, where *s* is the lowest-order or first expander index). The value must be a 22-character alphanumeric string.
firmware_level_4	Indicates the bootloader firmware level of the second lowest order expander index. (or *b+1*, where *b* is the lowest-order or first expander index). The value must be a 22-character alphanumeric string.
firmware_level_5	Indicates the complex programmable logic device (CPLD) firmware level of the lowest order expander index. The value must be a 22-character alphanumeric string.
firmware_level_6	Indicates the CPLD firmware level of the second lowest order expander index (or *c+1*, where *c* is the lowest-order or first expander index). The value must be a 22-character alphanumeric string.

A concise invocation example

```
lsenclosuresem 1
```

The resulting output:

```
enclosure_id sem_id status expander1_status expander2_status
1            1      online online           online
1            2      online online           online
```

A detailed invocation example

```
lsenclosuresem -sem 1 1
```

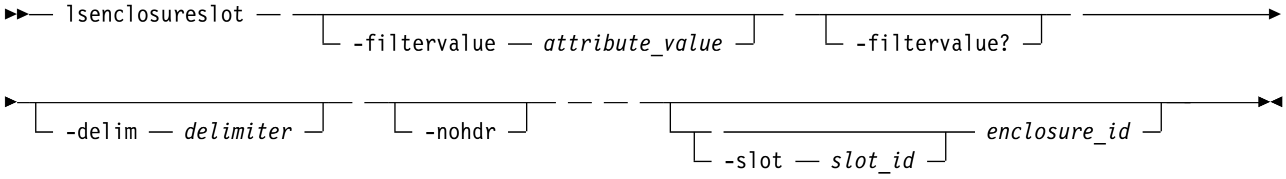
The resulting output:

```
enclosure_id 1
sem_id 1
status online
expander1_status online
expander2_status online
error_sequence_number
FRU_Part_Number *******
FRU_Identity **********************
firmware_level_1 0802.official
firmware_level_2 000E
firmware_level_3 0802.official
firmware_level_4 000E
firmware_level_5 1A.04.E3
firmware_level_6 1A.04.E5
```

lsenclosureslot

Use the **lsenclosureslot** command to view information about each drive slot in the enclosure.

Syntax



Parameters

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards:

- The wildcard character is the asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, enclose the filter entry within double quotation marks (""):

```
lsenclosureslot -filtervalue "enclosure_id>2"
```

-filtervalue?
(Optional) Displays the valid filter attributes for the **-filtervalue** parameter:

- drive_id
- drive_present
- enclosure_id
- port_1_status
- port_2_status
- slot_id

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each

column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (**:**) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. This parameter suppresses the display of these headings.

-slot *slot_id*
(Optional) Specifies the slot to display information for (it provides a detailed view for that enclosure slot). This parameter is valid only when an enclosure is specified. The value must be a number in the range 1 - 92.

Note:

If slot information is requested for a slot that does not exist on the specified enclosure, value that is displayed is blank.

enclosure_id
(Optional) Lists slots for that enclosure. Must be specified whether **-slot** is used.

Description

This command displays information about each drive slot in the enclosure, such as whether a drive is present, and the port status for that drive. Table 71 shows the possible outputs:

Table 71. lsenclosureslot output

Attribute	**Description**
enclosure_id	The identity of the enclosure that contains the drive slot.
slot_id	Identifies which of the drive slots in the enclosure it is.
port_1_status	The status of enclosure slot port 1. If the port is bypassed for multiple reasons, only one is shown. In order of priority, they are: • `online` indicates that the enclosure slot port 1 is online • `excluded_by_drive` indicates that the drive excluded the port • `excluded_by_enclosure` indicates that the enclosure excluded the port • `excluded_by_system` indicates that the clustered system (system) excludes the port
port_2_status	The status of enclosure slot port 2. If the port is bypassed for multiple reasons, only one is shown. In order of priority, they are: • `online` indicates that the enclosure slot port 2 is online • `excluded_by_drive` indicates that thethe drive excluded the port • `excluded_by_enclosure` indicates that thethe enclosure excluded the port • `excluded_by_system` indicates that thethe clustered system (system) excludes the port
fault_LED	The state of the combined fault and identify light-emitting diodes (LEDs): • `off` indicates no fault • `slow_flashing` indicates the `identify` mode **Note:** When the LED is in identify mode, it conceals whether there is a fault present (it always flashes). When you remove it from identity mode, the LED is turned `on` or `off`. • `on` indicates fault
powered	Indicates whether the slot is powered on. The values are `yes` or `no`.

Table 71. lsenclosureslot output (continued)

Attribute	Description
drive_present	Indicates if a drive is in the slot. The drive can be working, dead, or powered off. The values are yes (present) or no (empty).
drive_id	Indicates the ID of the drive in the slot; blank if there is no drive present, or if there is a drive present but it is offline and unmanaged.
error_sequence_number	Indicates the error log number of the highest priority error for this object. This value is typically blank. However, if there is a problem (for example, the status is degraded), then it contains the sequence number of that error.
interface_speed	Indicates the lowest interface speed for the connected drive slot (in gigabits per second, or Gbps). The values are: • 1.5 Gbps • 3 Gbps • 6 Gbps • 12 Gbps • Blank if both ports are isolated or the drive is not connected
row	Identifies the row in which the slot appears. The value must be a letter in the range A - G.
column	Identifies the column in which the slot appears. The value must be a number in the range 1 - 14.

A concise invocation example

This example displays information about mappings between the 1 and 2 dimensional IDs

```
lsenclosureslot
```

The resulting output:

```
enclosure_id slot_id port_1_status port_2_status drive_present drive_id row column
1            1       online        online        no                     A   1
1            2       online        online        no                     A   2
1            3       online        online        no                     A   3
1            4       online        online        no                     A   4
1            5       online        online        no                     A   5
...
1            87      online        online        no                     G   9
1            88      online        online        no                     G   10
1            89      online        online        no                     G   11
1            90      online        online        no                     G   12
1            91      online        online        no                     G   13
1            92      online        online        no                     G   14
```

A detailed invocation example showing slot 2 in enclosure 5

```
lsenclosureslot  -delim : -slot 2 5
```

The resulting output:

```
enclosure_id:5
slot_id:2
port_1_status:online
port_2_status:online
fault_LED:off
powered:yes
drive_present:yes
drive_id:105
error_sequence_number:
interface_speed:6Gb
```

A detailed invocation example

```
lsenclosureslot -delim :
```

The resulting output:

```
enclosure_id:slot_id:port_1_status:port_2_status:drive_present:drive_id:error_sequence_number
1:1:online:online:yes:22:
1:2:online:online:yes:23:
1:3:online:online:yes:19:
1:4:online:online:yes:7:
1:5:online:online:yes:10:
1:6:online:online:yes:18:
1:7:online:online:yes:20:
1:8:online:online:yes:16:
1:9:online:online:yes:12:
1:10:online:online:yes:11:
1:11:online:online:yes:21:
1:12:online:online:yes:9:
1:13:online:online:yes:14:
1:14:online:online:yes:5:
1:15:online:online:yes:15:
1:16:online:online:yes:13:
1:17:online:online:yes:6:
1:18:online:online:yes:17:
1:19:online:online:yes:4:
1:20:online:online:yes:1:
1:21:online:online:yes:8:
1:22:online:online:yes:0:
1:23:online:online:yes:3:
1:24:online:online:yes:2:
```

lsenclosurestats

Use the **lsenclosurestats** command to display the most recent values (averaged) of all enclosure statistics. It can also display a history of those values for any subset of the available statistics.

Syntax

```
►►── ── lsenclosurestats ── ┬─────────┬─ ┬────────────────┬─ ──────────────────────►
                            └─ -nohdr ┘  └─ -filtervalue? ┘

►─┬──────────────────────────────┬─ ┬────────────────────┬─ ──────────────────────►
  └─ -filtervalue ──attribute=value┘  └─ -delim ── delimiter ┘

►─┬──────────────────────────────────┬────────────────────────────────────────────►◄
  └─┬────────────────────┬─ ──enclosure_id┘
    └─ -history stat_list ┘
```

Parameters

-history stat_list
(Optional) Produces a history of values for enclosure statistics.

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-filtervalue *attribute=value*

(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards:

- The wildcard character is the asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, enclose the filter entry within double quotation marks (""):

```
lsenclosurestats -filtervalue "enclosure_id>2"
```

-filtervalue?

(Optional) Displays the valid filter attributes for the **-filtervalue** parameter:

- enclosure_id
- stat_name

-delim *delimiter*

(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

enclosure_id

(Optional) Indicates the unique enclosure identifier (a number in the range 1 - 99).

Description

Remember: This command cannot be used for products that do not support environmental statistics.

If you specify **-history stat_list** you must also specify *enclosure_id*. Filtering is supported for the concise view but not the detailed view.

Multiple statistical histories can be requested. The limit is the current maximum number of different statistical names published in the concise view. The concise view defines the output order.

For the detailed view, enclosure power is averaged over thirty seconds to provide immediate power.

Note: Averaging applies only to populated samples
Enclosure power is not averaged in the output if **-history** is specified.

The following is an invocation example for products that do not support environmental statistics - a message is displayed:

```
lsenclosurestats
```

This is the resulting output:

```
CMMVC6051E An unsupported action was selected.
```

Table 72 on page 427 displays information about chassis-specific enclosure properties and shows possible outputs for products that support environmental statistics.

Table 72. lsenclosurestats outputs

Attribute	Description
enclosure_id	Indicates the enclosure identifier; it can be a numeric character in the range 1 - 264.
sample_time	Indicates the time during which the sample occurred.
stat_name	Indicates the name of the statistical field.
stat_current	Indicates the current value of the statistical field.
stat_peak	Indicates the peak value of the statistic field. The last 5 minutes are used for samples.
stat_peak_time	Indicates the time that the peak occurred.
stat_value	Indicates the value of the statistic.

Remember: Filtering is supported on the `enclosure_id` and `stat_name` fields by using the concise view.

An invocation example

```
lsenclosurestats
```

The resulting output:

```
enclosure_id stat_name stat_current stat_peak stat_peak_time
1            power_w   2200         2500      120402103212
1            temp_c    35           36        120402103212
1            temp_f    95           97        120402103212
2            power_w   2300         2600      120402102917
2            temp_c    36           37        120402102917
2            temp_f    97           98        120402102917
4            power_w   2100         2400      120402103202
4            temp_c    33           35        120402103202
4            temp_f    93           95        120402103202
```

An invocation example

```
lsenclosurestats -history power_w 1
```

The resulting output:

```
enclosure_id sample_time  stat_name stat_value
1            120402105137 power_w   2282
1            120402105142 power_w   2290
1            120402105147 power_w   2281
1            120402105152 power_w   2290
1            120402105157 power_w   2281
1            120402105202 power_w   2289
1            120402105207 power_w   2282
1            120402105212 power_w   2289
1            120402105217 power_w   2281
1            120402105222 power_w   2289
1            120402105227 power_w   2281
1            120402105232 power_w   2290
1            120402105237 power_w   2282
1            120402105242 power_w   2289
1            120402105247 power_w   2282
1            120402105252 power_w   2289
1            120402105257 power_w   2282
1            120402105302 power_w   2289
1            120402105307 power_w   2282
1            120402105312 power_w   2289
1            120402105317 power_w   2282
1            120402105322 power_w   2287
1            120402105327 power_w   2281
```

```
1           120402105332 power_w   2290
1           120402105337 power_w   2281
1           120402105342 power_w   2289
1           120402105347 power_w   2282
1           120402105352 power_w   2289
1           120402105357 power_w   2281
1           120402105402 power_w   2289
1           120402105407 power_w   2281
1           120402105412 power_w   2289
1           120402105417 power_w   2282
1           120402105422 power_w   2289
1           120402105427 power_w   2282
1           120402105432 power_w   2289
1           120402105437 power_w   2281
1           120402105442 power_w   2290
1           120402105447 power_w   2281
1           120402105452 power_w   2290
1           120402105457 power_w   2282
1           120402105502 power_w   2287
1           120402105507 power_w   2281
1           120402105512 power_w   2290
1           120402105517 power_w   2281
1           120402105522 power_w   2289
1           120402105527 power_w   2282
1           120402105532 power_w   2290
1           120402105537 power_w   2281
1           120402105542 power_w   2290
1           120402105547 power_w   2281
1           120402105552 power_w   2290
1           120402105557 power_w   2281
1           120402105602 power_w   2289
1           120402105607 power_w   2282
1           120402105612 power_w   2289
1           120402105617 power_w   2281
1           120402105622 power_w   2289
1           120402105627 power_w   2281
1           120402105632 power_w   2290
```

This table provides the possible values that are applicable to the values that are displayed for the **stat_name** attribute.

Table 73. Stat_name field values

Value	Description
power_w	Displays the power that is consumed in watts.
temp_c	Displays the ambient temperature in Celsius.
temp_f	Displays the ambient temperature in Fahrenheit.

lssasfabric

Use the **lssasfabric** command to see which canisters are visible to a node, and the order of these canisters.

Syntax

```
►►─ lssasfabric ─┬──────────────────────────────────┬─┬──────────┬─────────────►◄
                 └─ -filtervalue ─ attribute_value ─┘ └─ -nohdr ─┘
```

```
►──┬──────────────────────┬── ┬─────────────────┬── ──────────────────────────────────────►◄
   └─ -delim ── delimiter ┘   └─ -filtervalue? ─┘
```

Parameters

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards:

- The wildcard character is the asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, enclose the filter entry within double quotation marks (""): `lssasfabric -filtervalue status`

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-filtervalue?
(Optional) Displays the valid filter attributes for the **-filtervalue** parameter:

- enclosure_id
- canister_id
- canister_port_id
- control_enclosure_id
- node_canister_id
- node_canister_port_id
- position
- IO_group_id
- IO_group_name
- node_id
- node_name

Description

Use this command to see which canisters are visible to a node, and the order of these canisters. Table 74 on page 430 describes possible outputs.

Table 74. lssasfabric output

Attribute	**Description**
enclosure_id	Indicates the identity of the enclosure the strand goes to.
canister_id	Indicates the canister in the enclosure that the strand goes to.
canister_port_id	Indicates the canister port that the strand goes to.
control_enclosure_id	Indicates the identity of the enclosure that the strand comes from. If the node does not reside inside a canister or an enclosure this field is blank.
node_canister_id	Indicates the identity of the canister the strand comes from. If the node does not reside inside a canister or an enclosure this field is blank.
node_canister_port_id	Indicates the node canister port that the strand is from. This must be the same as the chain ID.
position	Indicates the position in the strand or chain.
IO_group_id	Indicates the I/O group that the strand belongs to. This must be the same as the enclosure IO group.
IO_group_name	Indicates the I/O group the strand belongs to. This must be the same as the enclosure IO group.
node_id	Indicates the identity of the node that the strand is from. This is the same physical object as the node_canister
node_name	The name of the node that the strand is from. This is the same physical object as the node_canister.

An invocation example with three enclosures

Enclosure 1 is the control enclosure. Enclosure 2 is on chain 1 (node canister port 1) using canister port 1 as its connector. Enclosure 3 is on chain 2 (node canister port 2) using canister port 2 as its connector.

```
lssasfabric
```

Note: In this guide, the following output is split into two parts. This is for illustrative purposes; the output does not appear in two parts when you run this command.

This is the first part of the resulting output:

```
enclosure_id  canister_id  canister_port_id  control_enclosure_id   node_canister_id
           1            1                 1                     1                  1
           1            2                 1                     1                  2
           2            1                 1                     1                  1
           2            2                 1                     1                  2
           3            1                 2                     1                  1
           3            2                 2                     1                  2
```

This is the second part of the resulting output:

```
node_canister_port_id     position  IO_group_id  IO_group_name  node_id  node_name
                    2            0            0       io_grp0         1       node1
                    2            0            0       io_grp0         2       node2
                    1            1            0       io_grp0         1       node1
                    1            1            0       io_grp0         2       node2
                    2            1            0       io_grp0         1       node1
                    2            1            0       io_grp0         2       node2
```

An invocation example with two enclosures

This example shows the output when you use this command for a pair of expansion enclosures that are wired correctly to a set of nodes.

```
lssasfabric
```

The resulting output:

```
enclosure_id canister_id canister_port_id control_enclosure_id node_canister_id node_canister_port_id position IO_group_id IO_group_name node_id node_name
1            1           1                                                       1                     1        0           io_grp0       1       node1
2            1           1                                                       2                     1        0           io_grp0       1       node1
1            2           1                                                       1                     1        0           io_grp0       2       node2
2            2           1                                                       2                     1        0           io_grp0       2       node2
```

[edit]

resetleds

Use the **resetleds** command to simultaneously switch off all light-emitting diodes (LEDs) in the clustered system (system), including node battery LEDs.

Syntax

```
►►—resetleds——————————————————————————————————————————————————————————————————►◄
```

Parameters

None.

Description

The **resetleds** command simultaneously switches off all LEDs in the system, including node battery LEDs. This ensures that any identity LED switched on is the only one in the system that is switched on. This command works only on LEDs that are on systems that can communicate, which means they are online or of a supported type. This command fails if an object is offline, or if the enclosure is an unsupported type. This command does not affect LEDs:

- On independently-controlled objects
- On offline objects
- With hardware-only controls

An invocation example

```
resetleds
```

The resulting output:

```
No feedback
```

triggerenclosuredump

Use the **triggerenclosuredump** command to force the specified enclosure or enclosures to dump data.

Syntax

```
►►— triggerenclosuredump — ——┬— -port —port_id— -iogrp —iogrp_id_or_name—┬——————————————————►◄
                             └— -enclosure —enclosure_id——————————————————┘
```

Note:

1. You can only use one of the optional parameters (**-port** or **-enclosure**).
2. If **-port** is specified, **-iogrp** must also be specified.
3. If **-iogrp** is specified, **-port** must also be specified.

Parameters

-port *port_id*
(Optional) If the system is wired correctly, this value is identical to the ID of the chain with the enclosures you want to dump. If the system is wired incorrectly, all the enclosures connected to port *port_id* of either node canister are dumped.

-iogrp *iogrp_id_or_name*
(Optional) The ID or name of the I/O group the control enclosure belongs to.

-enclosure *enclosure_id*
(Optional) The ID of the enclosure you want to dump.

Description

Important: One of the optional parameters must be specified.

This command requests the canisters in the enclosure or enclosures specified to dump data. The dumped data is subsequently collected and moved to `/dumps/enclosure` on the nodes that are connected to the enclosure. There is one file for each canister successfully dumped and they may be located on different nodes. Dumps are for use by your prouct support team (or information) if it has the tools to interpret the dump data. Use the **cpdumps** command to copy the files from the system. This command does not disrupt access to the enclosures. The system limits the number of enclosure statesaves in the directory to 20 per node.

To trigger enclosure dumps from all enclosures connected to port 1 of the control enclosure in `iogrp 2`

```
triggerenclosuredump -port 1 -iogrp 2
```

The resulting output:

```
The data is dumped to the /dumps/enclosure directory if command is successful.
```

To trigger enclosure dumps from `enclosure 5`

```
triggerenclosuredump -enclosure 5
```

The resulting output:

```
The data is dumped to the /dumps/enclosure directory if command is successful.
```

Chapter 14. Encryption commands

Use the encryption and security commands are used to create, change, or list system encryption feature details.

chencryption

Use the **chencryption** command to manage the encryption state of the system.

Syntax

```
►►── chencryption ──┬──────────────────────────────────────────┬── ──────────────────────────►
                    └─ -usb ──┬─ enable ──────────────────────┤
                              ├─ disable ─────────────────────┤
                              ├─ validate ────────────────────┤
                              └─ newkey ── -key ──┬─ prepare ─┘
                                                  ├─ commit ──┤
                                                  └─ cancel ──┘

►─┬─────────────────────────────────────────┬────────────────────────────────────────────►◄
  └─ -keyserver ──┬─ enable ────────────────┤
                  ├─ disable ───────────────┤
                  └─ newkey ── -key ──┬─ prepare ─┘
                                      ├─ commit ──┤
                                      └─ cancel ──┘
```

Parameters

-usb *enable* | *disable* | *validate* | *newkey*

(Required if you do not specify **-keyserver**) Specifies whether USB encryption is enabled (or disabled) or the encryption keys are validated. You can also create new encryption keys that are also stored on Universal Serial Bus (USB) flash drives for use if the system forgets the encryption keys.

-usb *enable*

Enables encryption capability on the system. Then specify **-usb** *newkey* to create new keys. Use this command when the system has encryption hardware and encryption licenses (for example, the **lsencryption** value for `status` is set to `licensed`).

-usb *disable*

Disables the encryption capability of the system. If no encryption key is prepared this operation is complete and no further action is needed. Do not use this command if an encryption key is prepared or encrypted objects exist.

Remember: This removes all encryption keys (that are not on the USB flash drive) from the system.

-usb *validate*

Verifies that encryption keys are present on the USB flash drive and makes sure that the keys match the system encryption keys. Use this command when encryption is enabled and encryption keys exist (for example, **lsencryption** value for `usb_rekey` is set to `no`).

-usb *newkey*

Generates a new encryption key on a USB flash drive that is attached to the system. Use this command only if the minimum number of USB flash drives that can be used as key material stores are attached to the system (as reported by **lsportusb**). When you specify this parameter, the **-key** option must also be supplied.

-keyserver *enable* | *disable* | *newkey*
(Required if you do not specify **-usb**) Specifies the encryption task that involves encryption keys that are managed by key servers.

-keyserver *enable*
Enables encryption capability on the system. Use this command when the system has encryption hardware and encryption licenses (for example, the **lsencryption** value for `keyserver_status` is set to `licensed`).

-keyserver *disable*
Disables the encryption capability of the system. If no encryption key is prepared, this operation is complete and no further action is needed. Do not use this command if an encryption key is prepared or encrypted objects exist.

-keyserver *newkey*
Generates a new encryption key on the primary key server that is attached the system. You must also specify **-key** when you specify this parameter.

-key *prepare* | *commit* | *cancel*
(Optional) Manages the creation of a new or replacement (rekey) encryption keys when `-usb newkey` or `-keyserver newkey` is specified. There are three stages:

-key *prepare*
Generates system encryption keys and writes those keys to all system attached USB flash drives or key servers. If there is active encryption key material, confirm that at least one USB flash drive or key server has the current key material. Use this command only when the **lsencryption** value for `usb_rekey` or `keyserver_rekey` is set to `no` or `no_key`.

-key *commit*
Commits the prepared key as the current key. Use this command when the **lsencryption** value for `usb_rekey` or `keyserver_rekey` is set to `prepared` and the number of USB encryption keys is at least the minimum number required.

-key *cancel*
Cancels any specified key changes. Use this command when the **lsencryption**value for `usb_rekey` or `keyserver_rekey` is set to `prepared`.

Description

Use this command to manage the encryption state of the system. You must specify either **-usb** or **-keyserver**.

You can use this command can to turn on or turn off USB key encryption or key server encryption (but you cannot disable encryption if there are any encrypted objects). There are four types:

- `enable`, which enables encryption
- `disable`, which disables encryption
- `validate`, which validates encryption

  Note: The `validate` option does not apply to key server encryption.
- `newkey`, which specifies a new key for encryption

You can also perform a rekey of the external USB key or key server key material, which is divided into three stages:

- `prepare`, which generates new keys and sets up the system to change encryption keys during apply
- `commit`, which includes applying new keys (and copying key material)

- `cancel`, which rolls back the key setup that is performed during the `prepare` and cancels the rekey request

You cannot perform an enable, disable, or rekey operation for a key provider that is part of a cloud account that is in `import` mode.

You can use both USB flash drive and key server encryption in parallel on the same system. However, you must configure and administer these encryption methods independently.

An invocation example

```
chencryption -usb enable
```

The resulting output:

```
No feedback
```

An invocation example

```
chencryption -usb newkey -key prepare
```

The resulting output:

```
No feedback
```

An invocation example

```
chencryption -usb newkey -key commit
```

The resulting output:

```
No feedback
```

An invocation example

```
chencryption -keyserver enable
```

The resulting output:

```
chencryption -keyserver newkey -key prepare
```

An invocation example

```
chencryption -keyserver newkey -key commit
```

The resulting output:

```
No feedback
```

chkeyserver

Use the **chkeyserver** command to change the attributes for a key server object.

Syntax

```
►►— chkeyserver — ┬──────────────────────┬─ ┬───────────────────┬─ ──────────────────────────►
                  └─ -ip — ip_address ─┘    └─ -port — port ─┘

►┬──────────────────────────────────┬─ ┬───────────────┬─ ┬──────────┬─ ┬─────────────┬─ ──────────►
 └─ -sslcert — certificate_file ─┘    └─ -nosslcert ─┘    └─ -name ─┘    └─ -primary ─┘

►┬─ object_id ───┬──────────────────────────────────────────────────────────────────────►◄
 └─ object_name ─┘
```

Parameters

-ip *ip_address*
(Optional) Specifies the key server's IP address. The value must be in the form of a standard Internet Protocol version 4 (IPv4) or Internet Protocol version 6 (IPv6) address.

-port *port*
(Optional) Specifies the key server's TCP/IP port. The value must be a number 1 - 65535. The default value is the same as the default port used for key servers of the currently enabled type.

-sslcert *certificate_file*
(Optional) Specifies the key server's self-signed certificate. The value must be a file path string.

-nosslcert
(Optional) Specifies the removal of the key server self-signed certificate.

-name
(Optional) Specifies the key server object name. The value must be an alphanumeric string.

-primary
(Optional) Specifies the primary key server.

object_id | *object_name*
(Required) Specifies the object name or ID that you want to modify.

Description

This command changes the attributes for a key server object.

When a primary key server is configured, that key server must be defined before a rekey operation occurs. A primary object (such as a server) can be configured at any time when a defined primary server is present. A rekey operation without a defined primary key server fails.

An invocation example

```
chkeyserver -primary vardy2
```

The resulting output:

```
No feedback
```

An invocation example

```
chkeyserver -name zlatibr4
```

The resulting output:

```
No feedback
```

An invocation example

```
chkeyserver -sslcert /tmp/yourcert.pem 0
```

The resulting output:

```
No feedback
```

chkeyserverisklm

Use the **chkeyserverisklm** command to change the system-wide IBM Security Key Lifecycle Manager key server configuration.

Syntax

```
►►─ chkeyserverisklm ─ ──┬─────────────────────────────────────┬──┬────────────┬──────────────────►◄
                         ├─ -devicegroup ─ device_group ───────┤  └─ -disable ─┘
                         ├──┬─────────────────────────────────┬┤
                         │  ├─ -sslcert ─ certificate_file ─┤ ││
                         │  └─ -nosslcert ──────────────────┘ ││
                         └─ -enable ──────────────────────────┘
```

Parameters

-devicegroup *device_group*
: (Optional) Specifies a specific device group that the system uses with a key server. The value must be an alphanumeric string no more than 16 characters long.

  Note: The specified device name must begin with a letter (not a number) and cannot contain an underscore.

-sslcert *certificate_file*
: (Optional) Specifies the certificate authority (CA) certificate for the key server. This parameter cannot be used with **-nosslcert**. The value must specified in base64-encoded PEM format.

-nosslcert
: (Optional) Specifies that the CA certificate on the key server is deleted. This parameter cannot be used with **-sslcert**.

-enable
: (Optional) Enables the specified key server type.

-disable
: (Optional) Disables the specified key server type.

  Important: Do not specify **-disable** with other parameters.

Description

This command changes the system-wide IBM Security Key Lifecycle Manager key server configuration.

An invocation example

```
chkeyserverisklm -devicegroup JVAR_IBRA -sslcert /dumps/CA_certificate.pem -enable
```

The resulting output:

```
No feedback
```

An invocation example

```
chkeyserverisklm -nosslcert
```

The resulting output:

```
No feedback
```

lsencryption

Use the **lsencryption** command to display system encryption information.

Syntax

```
►►— lsencryption — ─┬─────────┬─ ─┬────────────────────────┬──────────────────────────────►◄
                    └─ -nohdr ─┘   └─ -delim — delimiter ─┘
```

Parameters

-nohdr

(Optional) By default, headings are displayed for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*

(Optional) In a detailed view, each item of data has its own row, and if the headings are displayed, the data is separated from the heading by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. In a detailed view, the data is separated from its heading by the specified delimiter.

Description

Use this command to display output that is related to the system encryption state.

Table 75 describes possible outputs.

Table 75. lsencryption output

Attribute	Value
status	Indicates the system encryption status. • `not_supported`, which indicates that the system has no supported encryption function. • `not_licensed`, which indicates that the system supports encryption but not all licenses are installed. • `licensed`, which indicates that the system has licenses that are installed for all encryption-capable hardware. • `enabled` , which indicates that system encryption is working and ready to create encrypted storage.
error_sequence_number	Indicates the event log sequence number of any problem that affects encryption. If there is no problem, it is blank.
usb_rekey	Indicates the state of the Universal Serial Bus (USB) rekey process. • `no`, which indicates that there is no rekey process ongoing, but keys exist. • `no_key`, which indicates that there is no rekey process and keys do not exist. • `prepared`, which indicates that a rekey process is active and the system prepares a new key that is waiting for this command to be issued: **chencryption -usb newkey -key commit**. • `committing`, which indicates that a commit is in progress.
usb_key_copies	Indicates the number of USB devices that prepared keys are written to.
usb_key_filename	Indicates the name of the file that contains the current encryption key.
usb_rekey_filename	Indicates the name of the file that contains the current prepared encryption key.

Table 75. lsencryption output (continued)

Attribute	Value
keyserver_status	Indicates the encryption status for key server encryption. The values are: • `not_supported`, which indicates that the system has no supported encryption function. • `not_licensed`, which indicates that the system supports encryption but not all licenses are installed. • `licensed`, which indicates that the system has licenses that are installed for all encryption-capable hardware. • `enabled` , which indicates that system encryption is working and ready to create encrypted storage.
keyserver_rekey	Indicates the state of the key server rekey process. The values are: • `no`, which indicates that there is no rekey process ongoing, but keys exist. • `no_key`, which indicates that there is no rekey process and keys do not exist. • `prepared`, which indicates that a rekey process is active and the system prepares a new key that is waiting for this command to be issued: **`chencryption -keyserver newkey -key commit`**. • `committing`, which indicates that a commit is in progress.
keyserver_pmk_uid	Indicates the UID for the key server.
keyserver_rekey_pmk_uid	Indicates the UID (after a rekey process) for the key server.

An invocation example for an encrypted system with no rekey

```
lsencryption
```

The resulting output:

```
status enabled
error_sequence_number
usb_rekey no
usb_key_copies 0
usb_key_filename
usb_rekey_filename
keyserver_status disabled
keyserver_rekey no_key
keyserver_pmk_uid
keyserver_rekey_pmk_uid
```

An invocation example for an encrypted system during the rekey

```
lsencryption
```

The resulting output:

```
status enabled
error_sequence_number
usb_rekey prepared
usb_key_copies 3
usb_key_filename
usb_rekey_filename encryptionkey_0000020061800028_0010030C00000007_Cluster_9.19.88.231
keyserver_status enabled
keyserver_rekey prepared
keyserver_pmk_uid
keyserver_rekey_pmk_uid KEY-1b9dcbe7-8b1c-401d-9bc2-1791534689fc
```

An invocation example for an encrypted system after the rekey completes

```
lsencryption
```

The resulting output:

```
status enabled
error_sequence_number
usb_rekey no
usb_key_copies 3
usb_key_filename encryptionkey_0000020061800028_0010030C00000007_Cluster_9.19.88.231
usb_rekey_filename
keyserver_status enabled
keyserver_rekey committing
keyserver_pmk_uid
keyserver_rekey_pmk_uid KEY-1a9hlfd8-8b1c-401d-9xy4-2948374653fc
```

lskeyserver

Use the **lskeyserver** command to display the key servers that are available to the clustered system (system).

Syntax

```
►►— lskeyserver — ──┬────────┬── ──┬────────────────────┬── ──┬───────────────┬─────────────►◄
                    └ -nohdr ┘     └ -delim — delimiter ┘     ├ object_id ────┤
                                                              └ object_name ──┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim : on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

object_id | *object_name*
(Optional) Specifies the object name or ID that you want to display details for.

Description

This command displays all key servers that are available to the system.

This table provides the attribute values that can be displayed as output view data.

Table 76. ~`lskeyserver output

Attribute	Description
id	Indicates the key server ID.
name	Indicates the key server name.

Table 76. ~`lskeyserver output (continued)

Attribute	Description
status	Indicates the key server status type. The values are: • `online` • `degraded` • `offline`
err_seq_num	Indicates the event log sequence number of the highest priority problem that affects the key server.
IP_address	Indicates the key server Internet Protocol (IP) address.
port	Indicates the key server TCP/IP port.
type	Indicates the key server type.
primary	Indicates whether the server is a primary server.
cert_set	Indicates whether a certificate exists for this key server object.
certificate	Indicates a human-readable description of the SSL certificate. The value reads `0 fields` if there is no certificate.

An invocation example

```
lskeyserver
```

The resulting output:

```
id name          status   IP_address              port   type        primary   cert_set
0  isklm_primary online   10.0.1.54               8709   isklm       yes       yes
1  isklm_backup  online   10.0.1.55               8709   isklm       no        yes
2  keyserver2    offline  0:0:0:0:0:ffff:a00:138  1234   isklm       no        no
3  keyserver3    offline  0:0:0:0:0:ffff:a00:139  1234   isklm       no        no
```

An invocation example

```
lskeyserver 0
```

The resulting output:

```
id 0
name keyserver0
status online
err_seq_num
IP_address 10.0.1.54
port 8709
type isklm
primary yes
certificate 0 fields
```

lskeyserverisklm

Use the **`lskeyserverisklm`** command to display the system-wide IBM Security Key Lifecycle Manager key server configuration.

Syntax

```
►►— lskeyserverisklm — ─┬──────────┬─ ─┬───────────────────┬─ ──────────────────────────►◄
                        └─ -nohdr ─┘   └─ -delim — delimiter ─┘
```

Parameters

-nohdr

(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If no data is to be displayed, headings are not displayed.

-delim *delimiter*

(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

Description

This command displays the system-wide IBM Security Key Lifecycle Manager key server configuration.

This table provides the attribute values that can be displayed as output view data.

Table 77. lskeyserverisklm output

Attribute	Description
status	Indicates the key server status type. The values are: • `disabled` • `enabled_inactive` • `prepared` • `enabled_active`
device_group	Indicates the device group. The value is a 16-character alphanumeric string.
certificate	Indicates a human readable description of the server SSL certificate that is generated by the system.

An invocation example

```
lskeyserverisklm
```

The detailed resulting output:

```
status enabled_active
device_group VARDY_SYSTEM
certificate 58 fields
   Data:
       Version: 3 (0x2)
       Serial Number: 1431938814 (0x5559a6fe)
   Signature Algorithm: sha256WithRSAEncryption
       Issuer: C=GB, L=Hursley, O=IBM, OU=SSG, CN=2145/emailAddress=support@ibm.com
       Validity
           Not Before: May 18 08:46:54 2015 GMT
           Not After : May 14 08:46:54 2030 GMT
       Subject: C=GB, L=Hursley, O=IBM, OU=SSG, CN=2145/emailAddress=support@ibm.com
       Subject Public Key Info:
           Public Key Algorithm: rsaEncryption
               Public-Key: (2048 bit)
               Modulus:
                   00:de:1c:70:c2:91:87:3c:6a:92:91:f7:d9:a3:5b:
                   05:e6:91:f1:87:c1:25:38:61:ad:4d:d9:26:19:7b:
                    9e:61:a5:fd:b1:d1:eb:d1:e4:a8:78:21:75:58:80:
                    4a:5c:dd:5e:6c:8b:1b:de:57:f9:d5:1f:71:92:3e:
                    78:d5:a4:75:1e:11:b2:62:18:52:0f:4d:32:a8:fd:
                    2b:16:4f:42:d1:d6:70:af:86:eb:fe:a1:ab:bc:66:
                    8a:44:bc:e0:36:53:77:96:2f:74:7d:95:33:79:c2:
                    59:5e:e1:43:50:da:43:25:c4:5d:3a:ac:d7:82:ad:
                    34:d5:ba:4c:52:4a:c0:81:3a:ad:e8:33:fe:4f:be:
                    e8:47:fa:5b:1f:dd:d8:9e:3b:44:a6:b6:b9:43:d2:
                    d4:45:8e:cb:5b:bb:10:5b:c9:30:68:2c:30:b6:e4:
                    ea:59:6d:a2:37:a7:13:77:28:1d:13:68:58:7b:dd:
                    90:d6:a8:81:7b:79:9f:1e:e4:a7:67:1b:7b:c5:b4:
                    90:dc:6b:d4:1f:7e:e9:e3:7b:ac:26:59:11:f1:99:
                    34:f0:6a:50:41:76:ad:a3:30:74:8f:8f:f5:ed:1e:
                    21:77:ff:51:90:1b:83:fb:04:f0:62:3d:71:17:a5:
                    ab:44:e8:bc:b0:82:0d:af:af:ae:68:5a:cf:e3:c8:
                    a9:53
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                87:66:33:16:61:7A:8E:CA:B4:BA:78:7B:56:56:8A:9D:C5:96:80:76
            X509v3 Authority Key Identifier:
                keyid:87:66:33:16:61:7A:8E:CA:B4:BA:78:7B:56:56:8A:9D:C5:96:80:76

    Signature Algorithm: sha256WithRSAEncryption
         56:b1:5d:59:11:ae:7b:6e:29:cc:1f:a8:75:77:d2:65:d6:88:
         75:8e:b9:cd:d6:71:ac:7e:89:8c:65:68:36:a8:28:97:88:36:
         42:da:a4:58:9b:c6:ce:c1:56:c9:0e:c5:ce:e7:01:74:d0:66:
         d0:4d:d3:0f:84:53:f6:e5:89:8e:44:6d:70:13:45:9c:21:91:
         50:f4:b0:b7:cc:cb:18:e8:d7:b3:38:b4:f5:5d:36:51:8c:7e:
         52:d4:24:0f:1f:2e:0a:b4:b6:9b:cb:23:43:6c:16:a2:a5:de:
         84:8a:0d:28:3c:d9:3d:5d:a4:52:44:28:90:98:a6:26:a9:c9:
         87:6c:27:3f:ef:09:5f:9d:0b:40:8d:07:64:ee:33:d9:40:47:
         98:02:10:58:2b:54:33:d9:37:69:d4:13:e6:0d:ec:46:26:b1:
         c1:c5:15:7c:8d:89:26:f7:95:d9:2f:d9:33:8c:f0:1a:dc:08:
         19:eb:18:16:51:30:a3:c0:ee:be:86:7d:3d:91:61:d5:99:bf:
         5e:19:b9:89:72:e1:4c:ea:5e:2b:90:ce:ce:75:83:e0:c9:14:
         83:21:21:e0:f8:28:94:90:71:e6:13:ca:97:8c:e3:58:b9:0c:
         62:03:e5:1c:1b:6c:dd:c3:60:48:d4:78:24:8e:22:34:78:32:
         fe:45:ee:36
```

```
-----BEGIN CERTIFICATE-----
MIIDzTCCArWgAwIBAgIEVVmm/jANBgkqhkiG9w0BAQsFADBqMQswCQYDVQQGEwJH
QjEQMA4GA1UEBwwHSHVyc2xleTEMMAoGA1UECgwDSUJNMQwwCgYDVQQLDANTU0cx
DTALBgNVBAMMBDIxNDUxHjAcBgkqhkiG9w0BCQEWD3N1cHBvcnRAaWJtLmNvbTAe
Fw0xNTA1MTgwODQ2NTRaFw0zMDA1MTQwODQ2NTRaMGoxCzAJBgNVBAYTAkdCMRAw
DgYDVQQHDAdIdXJzbGV5MQwwCgYDVQQKDANJQk0xDDAKBgNVBAsMA1NTRzENMAsG
A1UEAwwEMjE0NTEeMBwGCSqGSIb3DQEJARYPc3VwcG9ydEBpYm0uY29tMIIBIjAN
BgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA3hxwwpGHPGqSkffZo1sF5pHxh8El
OGGtTdkmGXueYaX9sdHr0eSoeCF1WIBKXN1ebIsb3lf51R9xkj541aR1HhGyYhhS
D00yqP0rFk9C0dZwr4br/qGrvGaKRLzgNlN3li90fZUzecJZXuFDUNpDJcRdOqzX
gq001bpMUkrAgTqt6DP+T77oR/pbH93YnjtEpra5Q9LURY7LW7sQW8kwaCwwtuTq
WW2iN6cTdygdE2hYe92Q1qiBe3mfHuSnZxt7xbSQ3GvUH37p43usJlkR8Zk08GpQ
QXatozB0j4/17R4hd/9RkBuD+wTwYj1xF6WrROi8sIINr6+uaFrP48ipUwIDAQAB
o3sweTAJBgNVHRMEAjAAMCwGCWCGSAGG+EIBDQQfFh1PcGVuU1NMIEdlbmVyYXRl
ZCBDZXJ0aWZpY2F0ZTAdBgNVHQ4EFgQUh2YzFmF6jsq0unh7VlaKncWWgHYwHwYD
VR0jBBgwFoAUh2YzFmF6jsq0unh7VlaKncWWgHYwDQYJKoZIhvcNAQELBQADggEB
AFaxXVkRrntuKcwfqHV30mXWiHWOuc3Wcax+iYxlaDaoKJeINkLapFibxs7BVskO
xc7nAXTQZtBN0w+EU/bliY5EbXATRZwhkVD0sLfMyxjo17M4tPVdNlGMflLUJA8f
Lgq0tpvLI0NsFqKl3oSKDSg82T1dpFJEKJCYpiapyYdsJz/vCV+dC0CNB2TuM9lA
R5gCEFgrVDPZN2nUE+YN7EYmscHFFXyNiSb3ldkv2TOM8BrcCBnrGBZRMKPA7r6G
fT2RYdWZv14ZuYly4UzqXiuQzs51g+DJFIMhIeD4KJSQceYTypeM41i5DGID5Rwb
bN3DYEjUeCSOIjR4Mv5F7jY=
-----END CERTIFICATE-----
```

mkkeyserver

Use the **mkkeyserver** command to create a key server object.

Syntax

```
►►— mkkeyserver — — -ip —ip_address— ┬──────────────┬ ┬───────────────────────────────┬──►
                                     └ -port —port─┘ └ -sslcert —certificate_file─┘

►– ┬─────────┬ ┬────────────┬──────────────────────────────────────────────────────────►◄
   └ -name ─┘ └ -primary ─┘
```

Parameters

-ip *ip_address*
(Required) Specifies the key server's IP address. The value must be in the form of a standard Internet Protocol version 4 (IPv4) or Internet Protocol version 6 (IPv6) address.

-port *port*
(Optional) Specifies the key server's TCP/IP port. The value must be a number 1 - 65535. The default value is the same as the default port used for key servers of the currently enabled type.

-sslcert *certificate_file*
(Optional) Specifies the key server's self-signed certificate. The value must be a file path string.

-name
(Optional) Specifies the key server object name. The value must be an alphanumeric string.

-primary
(Optional) Specifies the primary key server.

Description

This command creates a key server object.

The primary key server object is created by specifying **-primary**. If key management is enabled, you must use the primary key server object to create keys.

Note: When a primary key server is configured, that key server must be defined before a rekey operation occurs. A primary object (such as a server) can be configured at any time when a defined primary server
2 is present. When you create keys, the system uses the key server that is configured as the primary key
2 server. For multi-master key server configurations, any key server can be selected as the primary. A rekey operation without a defined primary key server fails.

When a key server object is created, it is automatically validated. If the validation is not successful, the command fails and an error message is displayed.

An invocation example

```
mkkeyserver -ip 10.0.1.54 -sslcert /tmp/isklm_public_server_cert.pem -primary
```

The resulting output:

```
Key Server, id [0], successfully created
```

An invocation example

```
mkkeyserver -ip 9.174.157.3 -name pogba_zibra -sslcert pogba_zibra_system_cert.pem
```

The resulting output:

```
Key Server, id [1], successfully created
```

rmkeyserver

Use the **rmkeyserver** command to remove a key server object.

Syntax

```
►►── rmkeyserver ── ─┬─ object_id ───┬──────────────────────────────────────────►◄
                     └─ object_name ─┘
```

Parameters

object_id | *object_name*
(Required) Specifies the object name or ID to remove.

Description

This command removes a key server object.

An invocation example

```
rmkeyserver 1
```

The resulting output:

```
No feedback
```

testkeyserver

Use the **testkeyserver** command to test key server objects.

Syntax

```
►►── testkeyserver ── ─┬─ object_id ───┬────────────────────────────────────────►◄
                       └─ object_name ─┘
```

Parameters

object_id | *object_name*
(Required) Specifies the object name or ID to be validated.

Description

This command tests key server objects.

An invocation example

```
testkeyserver 0
```

The resulting output:

```
The key server task completed successfully.
```

Chapter 15. Licensing and featurization commands

Use the licensing and featurization commands to work with licensed system functions.

activatefeature

Use the **activatefeature** command to activate a feature (using a license key or keyfile) or feature trial period.

Syntax

```
►►── activatefeature ── ─┬──────────────────────────────┬─ ─┬─────────────────────┬─ ──────────────►
                         └─ -trial ──┬─feature_id───┬─┘   └─ -licensekey ──key─┘
                                     └─feature_name─┘

►─┬───────────────────────────┬──────────────────────────────────────────────────────────────────►◄
  └─ -licensekeyfile ──filepath─┘
```

Parameters

-trial *feature_id* | *feature_name*
: (Optional) Activates the trial period for the feature of the specified ID that uses an unsigned 16-bit integer:

- Valid integer values are `0`, `1`, and `3`.
- Valid names are `turbo_performance`, `easy_tier`, and `remote_mirroring`.

-licensekey *key*
: (Optional) Provides the license key to activate a feature that contains 16 hexadecimal characters organized in four groups of four numbers with each group separated by a hyphen (such as `0123-4567-89AB-CDEF`).

-licensekeyfile *filepath*
: (Optional) Provides the full path-to-file containing all required license information by using an alphanumeric string that contains 1 - 256 characters.

Description

All parameters are mutually exclusive.

A license key file can contain one or more license keys. If you specify a key file, every key in the file is applied to the system. The license key is checked against the node or control enclosure serial number, machine type, and model. If no valid keys exist in the file, the command cannot complete successfully on the system. If you cannot apply a key successfully to the system, the command adds any remaining keys.

You must have one key for each node or control enclosure. Specify `activatefeature -licensekeyfile` with an `.xml` file that contains all node or control enclosure keys. Or, specify `activatefeature -licensekey` one time per node or control enclosure.

If a feature is already activated and you activate a feature again by using a key, the command completes successfully.

Remember:

- You cannot complete a trial when a feature is activated.

- You can activate a feature while a trial is in progress.

An invocation example

```
activatefeature -trial 1
```

The resulting output:

```
Activation of a trial is a one time operation.  Are you sure you wish to continue? Yes
```

An invocation example

```
activatefeature -licensekey 0123-4567-89AB-CDEF
```

The resulting output:

```
No feedback
```

An invocation example

```
activatefeature -licensekeyfile /tmp/keyfile.xml
```

The resulting output:

```
No feedback
```

chlicense

Use the **chlicense** command to change license settings for clustered system (system) features.

Syntax

This applies only to SAN Volume Controller :

```
►►— chlicense — ─┬─ -flash — capacity_TB ──────────────────────────────┬─┬──────────┬─────────────►◄
                 ├─ -remote — capacity_TB ─────────────────────────────┤ └─ -cloud ─┘
                 ├─ -virtualization — virtualization_setting_in_scu ───┤
                 ├─ -compression — compression_setting_in_scu ─────────┤
                 └─ -physical_flash ─┬─ on ──┬─────────────────────────┘
                                     └─ off ─┘
```

This applies only to Storwize V7000:

```
►►— chlicense — ─┬─ -remote ─────────┬────────────────────────────────────────────────────────────►◄
                 └─ -virtualization ─┘
```

This applies only to Storwize V5000.

```
►►— chlicense — ─┬─ -remote — enclosures ──────────────────────┬──────────────────────────────────►◄
                 ├─ -virtualization — number_of_enclosures ────┤
                 ├─ -easytier — enclosures ────────────────────┤
                 └─ -flash — enclosures ───────────────────────┘
```

Parameters

This applies to SAN Volume Controller , Storwize V7000:

-flash *capacity_TB*
(Optional) Changes system licensing for the FlashCopy feature. To change the licensed capacity for the FlashCopy feature, specify a capacity in terabytes (TB).

Note: Only use the optional **flash** parameter with the SAN Volume Controller.

-remote *capacity_TB*
(Optional) Changes system licensing for remote copy features such as Metro Mirror, Global Mirror, and HyperSwap. To change the licensed capacity for one of these features, specify a capacity in terabytes (TB). You must have an enclosure license for all enclosures.

Note: For Storwize V7000, specify the total number of internal and external enclosures that you have licensed on your system.

-virtualization *virtualization_setting_in_scu*
(Optional) Changes system licensing for the Virtualization feature. To change the licensed capacity for the Virtualization feature, specify a capacity in terabytes (TB).

Note: For Storwize V7000, specify the total number of internal and external enclosures that you have licensed on your system.

-physical_flash on | off
(Optional) For physical disk licensing, enables or disables the FlashCopy feature. The default value is **off**.

-compression *virtualization_setting_in_scu*
(Optional) Changes system licensing for the compression feature.

Note: All Storwize V7000 systems support compression.

To change the compression license capacity, specify a capacity value in terabytes (TB).

-cloud *enclosures*
(Optional) Specifies number of enclosures for the transparent cloud tiering feature. The value must be a number.

This applies only to Storwize V5000.

-remote *enclosures*
(Optional) Changes system licensing for the Metro Mirror, Global Mirror, or HyperSwap feature, and specifies the total number of internal and external enclosures that you have licensed on your system. You must have a Metro and Global Mirror enclosure license for all enclosures.

-virtualization *enclosures*
(Optional) Changes system licensing for the Virtualization feature. Specify the number of enclosures of external storage that you have been authorized to use.

-easytier *enclosures*
(Optional) Specifies the enclosures on which the customer can run Easy Tier.

-flash *enclosures*
(Optional) Specifies the total number of internal and external enclosures for the FlashCopy feature.

-compression *enclosures*
(Optional) Changes system licensing for the compression feature.

Description

The **chlicense** command changes license settings for the system. Any change that is made is logged as an event in the license setting log.

The capacity for each licensed feature can be modified with this command. This is the number of terabytes (TB) of volume capacity or Storage Control Units (SCU) capacity that can be configured by the system.

The enclosure license already includes virtualization of internal drives on your system. You can use this command to set any additional options. The total number of enclosures in your system must not exceed the total number of licensed enclosures that you have. The total virtualized capacity (number of external enclosures that can be configured by the system) can also be modified with this command. The default is to have no feature that is licensed, but it does not stop you from using related functions.

Any error that is placed in the license settings log results in a generic error being placed in the system error log. The command-line tool return code also notifies you that you are using an unlicensed feature.

When you reach 90% capacity, any attempt to create or extend volumes, relationships, or mappings generates an error message. You can still create and expand volumes, relationships, or mappings. When usage reaches or exceeds 100% capacity, errors are placed in the license settings log stating that you are using an unlicensed feature.

An invocation example for adding a remote copy license capacity of 5 TB

```
chlicense -remote 5
```

The resulting output:

```
No feedback
```

An invocation example for enabling Easy Tier settings

```
chlicense -easytier 2
```

The resulting output:

```
No feedback
```

An invocation example for modifying a compression license value

```
chlicense -compression 4
```

The resulting output:

```
No feedback
```

An invocation example for changing the license on a cloud account

```
chlicense -cloud 2
```

The resulting output:

```
No feedback
```

deactivatefeature

Use the **deactivatefeature** command to deactivate a feature or suspend a feature trial period.

Syntax

```
►►— deactivatefeature — —feature_id——————————————————————————————————————►◄
```

Parameters

feature_id
(Required) Deactivates the feature (or feature trial). This ID is the unique ID as displayed when you use the **lsfeature** command, and is an incremental number (in the range 0 - 320).

Description

Use this command to deactivate a feature or suspend a feature trial period.

An invocation example

```
deactivatefeature 1
```

The following output is displayed:

```
You are removing the ability to use a feature of this system. Are you sure you wish to continue?  Y
```

lsfeature

Use the **lsfeature** command to list the features that are available for the current clustered system (system) code release. You can also list trial or entitlement information and license keys.

Syntax

```
►►── lsfeature ─┬──────────────────────┬─ ─┬──────────┬─ ─┬──────────┬──────────────────►◄
                └─ -delim ── delimiter ─┘   └─ -nohdr ─┘   └─ -bytes ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim : on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-bytes
(Optional) Displays all capacities as bytes.

Description

This command lists the features that are available for the current clustered system (system) code release. You can also list trial or entitlement information and license keys.

This table provides the attribute values that can be displayed as output view data.

Table 78. lsfeature outputs

Attribute	Possible Values
id	Indicates the unique ID (2-character) feature number.
name	Indicates the feature name with a 16-character alphanumeric string: • `easy_tier` • `remote_mirroring` • `flashcopy_upgrade` • `turbo_performance` • `encryption`
state	Indicates the current state of the feature: • `active` • `inactive` • `trial_available` • `trial_active` • `trial_expired`
license_key	Indicates the key that is used in feature activation with a string that consists of 16 hexadecimal characters that are organized in four groups of four numbers with each group that is separated by a hyphen (such as `0123-4567-89AB-CDEF`).
trial_expiration_date	Indicates the trial expiration date as long as the state is `trial_available` or `trial_active`. This value is displayed in the format `YYYYMMDD`.
serial_num	Indicates the product serial number.
mtm	Indicates the machine type and model.

Note: A license key's association with an enclosure can be determined if:

- The enclosure that is associated with the key contains at least one node that is added to the cluster and the node is either online of offline at the time the view is queried.
- The enclosure that is associated with the key contains at least one node that is a candidate for clustered system membership. The node must be online at the time the view is being queried.

An invocation example

```
lsfeature
```

The resulting output:

```
id name              state           license_key         trial_expiration_date   serial_num mtm
0  turbo_performance trial_available                     20130201
1  easy_tier         trial_active                        20130101
2  flashcopy_upgrade active          0123-4567-89AB-CDEF
3  remote_mirroring  trial_expired                       20130201
```

An invocation example

In this system, both licenses are for encryption. There are two control enclosures, and the serial number and machine type are displayed:

```
lsfeature
```

The resulting output:

```
id name       state  license_key         trial_expiration_date serial_num mtm
0  encryption active 90AB-D41D-C799-2EF4                       78G00TT    2076-624
1  encryption active 3A87-463E-B5DF-9969                       31G00KG    2076-624
```

If one of the licenses is removed:

```
lsfeature
```

The resulting output:

```
id name       state    license_key         trial_expiration_date serial_num mtm
0  encryption inactive 90AB-D41D-C799-2EF4                       78G00TT    2076-624
```

The state is `inactive` because control enclosures require their own license to activate encryption.

An invocation example

In this system, both licenses are for encryption. There are two control enclosures, and the serial number and machine type are displayed:

```
lsfeature
```

The resulting output:

```
id name       state  license_key         trial_expiration_date serial_num mtm
0  encryption active 90AB-D41D-C799-2EF4                       78G00TT    2076-624
1  encryption active 3A87-463E-B5DF-9969                       31G00KG    2076-624
```

If one of the licenses is removed:

```
lsfeature
```

The resulting output:

```
id name       state    license_key         trial_expiration_date serial_num mtm
0  encryption inactive 90AB-D41D-C799-2EF4                       78G00TT    2076-624
```

The state is `inactive` because control enclosures require their own license to activate encryption.

lslicense

Use the **`lslicense`** command to display current license settings for clustered system (system) features.

Syntax

```
►►── lslicense ── ─┬──────────┬─ ─┬──────────────────────┬──────────────────────────►◄
                   └─ -nohdr ─┘   └─ -delim ── delimiter ─┘
```

Parameters

-nohdr
: (Optional) Suppresses the display of these headings. By default, headings are displayed for each column of data (in a concise style view that provides general information about objects of a particular type) and for each item of data (in a detailed style view that provides much more information about a specific object of a particular type).

  Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
: (Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The

-delim parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim :, a colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

Description

The **lslicense** command displays license settings for system features, including remote copy and virtualization settings.

SAN Volume Controller also includes FlashCopy settings. The displayed output for SAN Volume Controller lists capacity values in terabytes (TB) and feature enablement. The displayed output for Storwize V7000 lists enclosure license values.

Use the **chlicense** command to change the feature license settings. Because the feature license settings are entered when the system is first created, you must update the settings if you change your license.

Table 79 provides the possible values that are applicable to the attributes that are displayed as data in the output views.

Table 79. lslicense output

Attribute	**Possible Values**
used_flash	Indicates the amount Flash Copy (FC) memory used.
used_remote	Indicates the amount of remote copy memory used.
used_virtualization	Indicates the amount of virtualization memory used.
license_flash	Indicates the FC license settings.
license_remote	Indicates remote copy license settings.
license_virtualization	Indicates license virtualization settings.
license_physical_disks	Indicates the amount of physical disk space available for the license.
license_physical_flash	Indicates whether the license physical flash is on or off.
license_physical_remote	Indicates whether the physical remote copy license is on or off.
used_compression_capacity	Indicates the total virtual size of volumes with compressed copies, in total bytes (numeric format with two decimal places).
license_compression_capacity	Indicates the licensed compression capacity, in total bytes (numeric format).
license_compression_enclosures	Indicates which licensed enclosures have compression (numeric format).
license_easy_tier	Indicates which enclosures Easy Tier can be run on.
license_cloud_enclosures	Indicates whether a separate cloud account system storage license is configured.
scu_ratio_ssd	Indicates the storage capacity unit (SCU) ratio for SSD tier storage. The value must be a number with two decimal places.
scu_ratio_enterprise	Indicates the SCU ratio for enterprise tier storage. The value must be a number with two decimal places.
scu_ratio_nearline	Indicates the SCU ration for nearline tier storage. The value must be a number with two decimal places.

An invocation example

```
lslicense
```

The resulting output:

```
used_flash 0.00
used_remote 0.00
used_virtualization 0.00
license_flash 0
license_remote 20
license_virtualization 30
license_physical_disks 0
license_physical_flash on
license_physical_remote off
used_compression_capacity 0.02
license_compression_capacity 0
license_compression_enclosures 1


license_cloud_enclosures 0
scu_ratio_ssd 1.00
scu_ratio_enterprise 1.18
scu_ratio_nearline 4.00
```

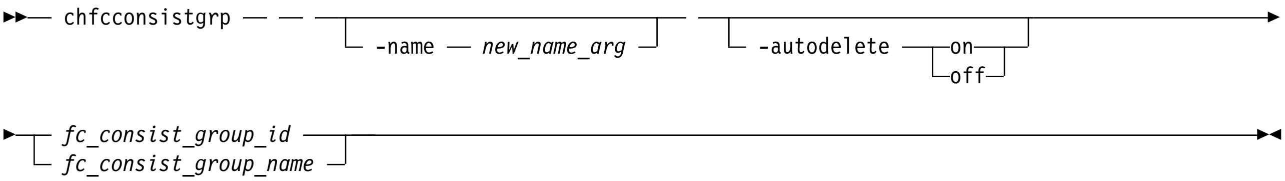
Chapter 16. FlashCopy commands

Use the FlashCopy commands to work with FlashCopy system methods and functions.

chfcconsistgrp

Use the **chfcconsistgrp** command to change the name of a consistency group or marks the group for auto-deletion.

Syntax



Parameters

-name *new_name_arg*
(Optional) Specifies the new name to assign to the consistency group.

-autodelete on | off
(Optional) Deletes the consistency group when the last mapping that it contains is deleted or removed from the consistency group.

fc_consist_group_id | fc_consist_group_name
(Required) Specifies the ID or existing name of the consistency group that you want to modify.

Description

The **chfcconsistgrp** command changes the name of a consistency group, marks the group for auto-deletion, or both.

Note: Maps that are `rc_controlled` are not shown in the view when this command is specified.

An invocation example

```
chfcconsistgrp -name testgrp1 fcconsistgrp1
```

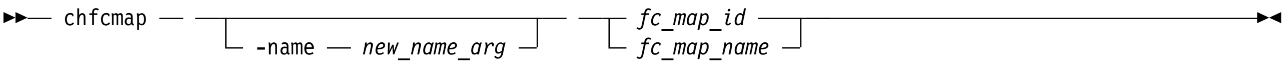
The resulting output:

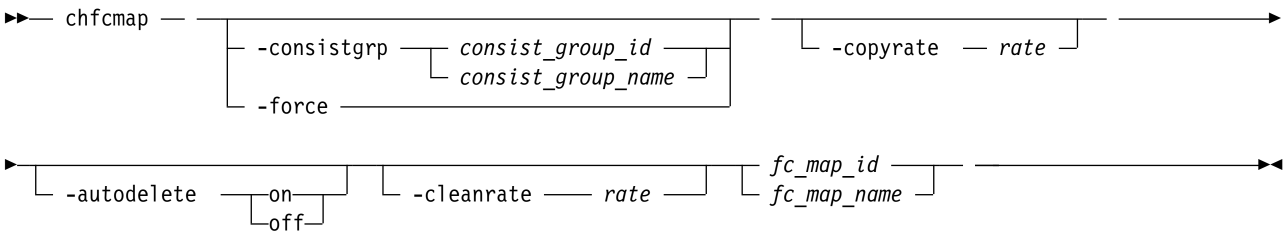
```
No feedback
```

chfcmap

Use the **chfcmap** command to modify attributes of an existing mapping.

Syntax





Parameters

-name *new_name_arg*
(Optional) Specifies the new name to assign to the mapping. The **-name** parameter cannot be used with any other optional parameters.

-force
(Optional) Specifies that the mapping be modified to a stand-alone mapping (equivalent to creating the mapping without a consistency group ID). You cannot specify the **-force** parameter with the **-consistgrp** parameter.

Important: Using the force parameter might result in a loss of access. Use it only under the direction of your product support information.

-consistgrp *consist_group_id* | *consist_group_name*
(Optional) Specifies the consistency group for which you want to modify the mapping. You cannot specify the **-consistgrp** parameter with the **-force** parameter.

Note: The consistency group cannot be modified if the specified consistency group is in the `preparing`, `prepared`, `copying`, `suspended`, or `stopping` state.

-copyrate *rate*
(Optional) Specifies the copy rate. The *rate* value can be 0 - 150. The default value is `50`. A value of `0` indicates no background copy process. For the supported **-copyrate** values and their corresponding rates, see Table 80 on page 459.

-autodelete on | **off**
(Optional) Specifies that the autodelete function be turned on or off for the specified mapping. When you specify the **-autodelete on** parameter, you are deleting a mapping after the background copy completes. If the background copy is already complete, the mapping is deleted immediately.

-cleanrate *rate*
(Optional) Sets the cleaning rate for the mapping. The *rate* value can be `0 - 150`. The default value is `50`.

fc_map_id | *fc_map_name*
(Required) Specifies the ID or name of the mapping to modify. Enter the ID or name last on the command line.

Description

The **chfcmap** command modifies attributes of an existing mapping.

Attention: You must enter the *fc_map_id* | *fc_map_name* last on the command line.

If you have created several FlashCopy mappings for a group of volumes that contain elements of data for the same application, you can assign these mappings to a single FlashCopy consistency group. You can then issue a single prepare command and a single start command for the whole group, for example, so that all of the files for a particular database are copied at the same time.

The **-copyrate** parameter specifies the copy rate. If 0 is specified, background copy is disabled. The **-cleanrate** parameter specifies the rate for cleaning the target volume. The cleaning process is only active if the mapping is in the copying state and the background copy has completed, the mapping is in the copying state and the background copy is disabled, or the mapping is in the stopping state. You can disable cleaning when the mapping is in the copying state by setting the **-cleanrate** parameter to 0. If the **-cleanrate** is set to 0, the cleaning process runs at the default rate of 50 when the mapping is in the stopping state to ensure that the stop operation completes.

This table provides the relationship of the copy *rate* and cleaning *rate* values to the attempted number of grains to be split per second. A grain is the unit of data represented by a single bit.

Table 80. Relationship between the rate, data rate, and grains per second values

User-specified *rate* attribute value	Data copied/sec	256 KB grains/sec	64 KB grains/sec
1 - 10	128 KB	0.5	2
11 - 20	256 KB	1	4
21 - 30	512 KB	2	8
31 - 40	1 MB	4	16
41 - 50	2 MB	8	32
51 - 60	4 MB	16	64
61 - 70	8 MB	32	128
71 - 80	16 MB	64	256
81 - 90	32 MB	128	512
91 - 100	64 MB	256	1024
101 - 110	128 MB	512	2048
111 - 120	256 MB	1024	4096
121 - 130	512 MB	2048	8192
131 - 140	1 GB	4096	16384
141 - 150	2 GB	8192	32768

Note: Maps that are rc_controlled are not shown in the view when this command is specified.

An invocation example

```
chfcmap -name testmap 1
```

The resulting output:

```
No feedback
```

lsfcconsistgrp

Use the **lsfcconsistgrp** command to display a concise list or a detailed view of FlashCopy consistency groups that are visible to the clustered system (system). This information is useful for tracking FlashCopy consistency groups.

The list report style can be used to obtain two styles of report:

- A list that contains concise information about all of the FlashCopy consistency groups on a system. (Each entry in the list corresponds to a single FlashCopy consistency group.)
- The detailed information about a single FlashCopy consistency group.

Syntax

```
►►── lsfcconsistgrp ──┬─────────────────────────────────┬──┬──────────┬──────────────►
                      └─ -filtervalue ── attribute=value ┘  └─ -nohdr ─┘

►─┬──────────────────────┬── ──┬─────────────────┬─┬───────────────┬─────────────────►◄
  └─ -delim ── delimiter ┘     └─ -filtervalue? ─┘ ├─ object_id ───┤
                                                   └─ object_name ─┘
```

Parameters

-filtervalue *attribute=value*

(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are returned.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards with the SAN Volume Controller CLI:

- The wildcard character is an asterisk character (*).
- The command can contain a maximum of one wildcard, which must be the first or last character in the string.
- When you use a wildcard, surround the filter entry with double quotation marks (""), as follows:

```
lsfcconsistgrp -filtervalue "name=md*"
```

-nohdr

(Optional) By default, headings are displayed for each item of data in a concise view. The **-nohdr** parameter suppresses the display of these headings. Detailed view is not valid for this command.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*

(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, the headers are displayed, and the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a o1-byte character. If you enter `-delim :` on the command line, the colon character (:) separates all items of data in a concise view; the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

object_id | *object_name*

(Optional) Specifies the name or ID of an object. When you use this parameter, the detailed view of the specific object is returned and any value that is specified by the **-filtervalue** parameter returns an error message. If you do not specify the *object_id* or *object_name* parameter, the concise view of all objects that match the filtering requirements that are specified by the **-filtervalue** parameter are displayed.

-filtervalue?

(Optional) Displays the list of valid filter attributes in the report. The valid filter attributes for the **lsfcconsistgrp** command are:

- `name`
- `id`
- `status`
- `FC_group_id`

Description

This command returns a concise list or a detailed view of FlashCopy consistency groups that are visible to the system.

The following list provides values of the `status` attribute that are displayed as data in the output views:

status Indicates the status. The values are:
- `idle_or_copied`
- `preparing`
- `prepared`
- `copying`
- `stopped`
- `suspended`
- `stopping`
- Empty

id Indicates the mapping ID.

name Indicates the mapping name.

start_time
Indicates the time that the group was started in `YYMMDDHHMMSS` format (or blank).

autodelete
Indicates whether auto deletion is `on` or `off`.

FC_mapping_id
Indicates the FlashCopy mapping ID.

FC_mapping_name
Indicates the FlashCopy mapping name.

A concise invocation example

```
lsfcconsistgrp -delim :
```

The concise resulting output:

```
id:name:status:start_time
1:ffccg0:empty:060627083237
2:ffccg1:idle_or_copied:060627083337
3:ffccg2:idle_or_copied:060627083437
```

A detailed invocation example

```
lsfcconsistgrp -delim : 1
```

The detailed resulting output:

```
id:1
name:ffccg0
status:empty
```

A detailed invocation example

```
lsfcconsistgrp -delim : fccstgrp0
```

The detailed resulting output:

```
id:1
name:FCcgrp0
status:idle_or_copied
start_time:060627083137
autodelete:off
```

```
FC_mapping_id:0
FC_mapping_name:fcmap0
FC_mapping_id:1
FC_mapping_name:fcmap1
```

lsfcmap

Use the **lsfcmap** command to generate a list that contains concise information about all of the FlashCopy mappings that are visible to the clustered system (system), or detailed information for a single FlashCopy mapping.

Syntax

```
►►— lsfcmap — ─┬──────────────────────────────────┬─ ─┬─────────┬─ ──────────────────────────►
               └─ -filtervalue — attribute=value ─┘   └─ -nohdr ─┘

►─┬─────────────────────────┬─ — ─┬────────────────┬─┬─────────────────┬──────────────────────►◄
  └─ -delim — delimiter ────┘     └─ -filtervalue? ─┘ ├─ object_id ─────┤
                                                      └─ object_name ───┘
```

Parameters

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards with the SAN Volume Controller CLI:

- The wildcard character is the asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, enclose the filter entry within double quotation marks (""), as follows:

  ```
  lsfcmap -filtervalue "name=md*"
  ```

-filtervalue?
(Optional) Displays the valid filter attributes for the **-filtervalue** *attribute=value* parameter:

- name
- id
- source_vdisk_id
- source_vdisk_name
- target_vdisk_id
- target_vdisk_name
- group_name
- group_id
- status
- copy_rate
- FC_mapping_name
- FC_id
- partner_FC_id
- partner_FC_name
- restoring

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed the data is separated from the header by a space. The **delim** parameter overrides this behavior. Valid input for the **delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (**:**) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

object_id | *object_name*
(Optional) Specifies the name or ID of an object. When you use this parameter, the detailed view of the specific object is returned and any value that is specified by the **filtervalue** parameter is ignored. If you do not specify the **object_ID** or **object_name** parameter, the concise view of all objects that match the filtering requirements that are specified by the **filtervalue** parameter are displayed.

Description

This command returns a concise list or a detailed view of FlashCopy mappings that are visible to the system.

The following list shows attribute values that can be displayed as output view data:

id Displays the mapping ID.

name Displays the mapping name.

source_vdisk_id
Displays the source volume ID.

source_vdisk_name
Displays the source volume name.

target_vdisk_id
Displays the target volume ID.

target_vdisk_name
Displays the target volume name.

group_id
Displays the group ID.

group_name
Displays the group name.

status Displays the status:

- `idle_or_copied`
- `preparing`
- `prepared`
- `copying`
- `stopped`
- `suspended`
- `stopping`

progress
Displays the progress.

copy_rate
Displays the copy rate.

start_time
Displays the time that the copy was last started. It is in the format `YYMMDDHHMMSS`. If a copy is not started, a blank line is displayed.

dependent_mappings
Displays any dependent mappings.

autodelete
Specifies if autodelete is on or off.

clean_progress
Indicates the clean progress.

clean_rate
Indicates the clean rate.

incremental
Indicates whether incremental is `on` or `off`.

difference
Indicates the difference.

IO_group
Displays the I/O group ID.

IO_group_name
Displays the I/O group name.

partner_FC_id
Displays the partner FlashCopy ID/

partner_FC_name
Displays the partner FlashCopy name.

restoring
Displays the restoring status. The values are `yes` or `no`.

rc_controlled
Displays the `rc_controlled` status.

copy_rate_mb
Displays the copy rate MB amount.

clean_rate_mb
Displays the clean rate MB amount.

keep_target
Displays the target and source volume availability. The values are:

- `yes` indicates that the source volume availability is connected to the availability of the target volume.
- `no` indicates that if there is a problem with the target volume that can impact FlashCopy operations, the target volume is removed.

restore_progress
Displays the percentage of the source volume that is restored from the target.

Note: Using `rc_controlled` indicates that the map is for internal use only. It cannot be manipulated externally.

A concise invocation example

```
lsfcmap -delim :
```

The concise resulting output:

```
id name source_vdisk_id:source_vdisk_name:target_vdisk_id:target_vdisk_name:group_id
   group_name:status:progress:copy_rate:clean_progress:incremental:partner_FC_id:
   partner_FC_name:restoring:start_time:rc_controlled
0:test:0:vdisk0:1:vdisk1:idle_or_copied:0:50:100:off:no
no0:fcmap0:0:vdisk0:1:vdisk1:0:fccstgrp0:idle_or_copied:0:50:0:on:2:fcmap2:no
1:fcmap1:2:vdisk2:3:vdisk3:0:fccstgrp0:idle_or_copied:0:0:100:off:::no
2:fcmap2:1:vdisk1:0:vdisk0:0:fccstgrp1:idle_or_copied:0:0:100:off:0:fcmap0:no
```

A detailed invocation example

```
lsfcmap 0
```

The detailed resulting output:

```
id:0
name:fcmap0
source_vdisk_id:63
source_vdisk_name:vdisk63
target_vdisk_id:57
target_vdisk_name:vdisk57
group_id:
group_name:
status:idle_or_copied
progress:0
copy_rate:0
start_time:
dependent_mappings:0
autodelete:off
clean_progress:100
clean_rate:50
incremental:off
difference:100
grain_size:256
IO_group_id:1
IO_group_name:io_grp1
partner_FC_id:
partner_FC_name:
restoring:no
rc_controlled:no
keep_target:yes
restore_progress:
```

lsfcmapcandidate

Use the **lsfcmapcandidate** command to list all of the volumes that are associated with fewer than 256 FlashCopy mappings.

Syntax

```
►►── ── lsfcmapcandidate ── ─┬──────────┬─ ─┬───────────────────────┬──────────────────►◄
                             └─ -nohdr ─┘   └─ -delim ── delimiter ─┘
```

Parameters

-nohdr

(Optional) By default, the heading is displayed for the column of data in a concise style view, and for the item of data in a detailed style view. The **-nohdr** parameter suppresses the display of the heading.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter **-delim :** on the command line, a colon character (**:**) separates all items of data in a concise view; the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

Description

This command returns a list of volumes that are associated with fewer than 256 FlashCopy mappings.

An invocation example

```
lsfcmapcandidate
```

The resulting output:

```
id
2
3
4
```

lsfcmapprogress

Use the **lsfcmapprogress** command to display the progress of the background copy of a FlashCopy mapping. This information is displayed as a percentage-completed value.

Syntax

```
►►─ lsfcmapprogress ─ ─┬─────────┬─ ─┬─────────────────────┬─ ─┬─ fcmap_id ───┬──────────────►◄
                       └─ -nohdr ─┘   └─ -delim ─ delimiter ─┘   └─ fcmap_name ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim : on the command line, the data is separated from its header by a colon character (**:**).

fcmap_id | *fcmap_name*
(Required) Specifies that you want the report to display the progress of the background copy for the designated FlashCopy mapping.

Description

This command reports a percentage for the progress of the background copy being copied on the specified FlashCopy mapping.

An invocation example

```
lsfcmapprogress 0
```

The resulting output:

```
id             progress
0              0
```

lsfcmapdependentmaps

Use the **lsfcmapdependentmaps** command to display the FlashCopy mappings that depend on the user specified mapping.

Syntax

```
►►── lsfcmapdependentmaps ── ─┬─────────┬─ ─┬────────────────────────┬─ ─┬─ fc_id ───┬──────────►◄
                              └─ -nohdr ─┘  └─ -delim ── delimiter ─┘  └─ fc_name ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

fc_id | *fc_name*
(Required) Specifies the name or ID of the FlashCopy mapping to list the dependent maps for.

Description

This command returns a list of dependent FlashCopy mappings. This command can be used to determine the list of FlashCopy mappings that would also stop if you stopped a mapping by using the **-force** parmeter.

There is a dependent_mapping_count field in the FlashCopy map detailed view (displayed when you process the **lsfcmap** command) that you can use as an indicator whether there are any dependent mappings in progress. If the count is zero, there are no dependent copies.

Note: If time elapses between the time you process the **lsfcmap** command and the **lsfcmapdependentmaps** command, there could be a difference between the actual number of dependent mappings that are being processed and the number that was reported by the **lsfcmap** command.

An invocation example

```
lsfcmapdependentmaps -delim : 2
```

The resulting output:

```
fc_id:fc_name
1:fcmap1
3:fcmap3
```

lsrmvdiskdependentmaps

Use the **lsrmvdiskdependentmaps** command to display all FlashCopy mappings that must be stopped for the specified volume to be deleted.

Syntax

```
►►— lsrmvdiskdependentmaps — ┬─────────┬─ ┬─────────────────────┬─ ─┬─ vdisk_name ─┬─────────►◄
                             └─ -nohdr ─┘ └─ -delim — delimiter ─┘   └─ vdisk_id ───┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim : on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

vdisk_name | *vdisk_id*
(Required) Specifies the name or ID of the volume for which the FlashCopy mappings are displayed.

Description

This command returns a list of the FlashCopy mappings that must be stopped before the specified volume can be deleted. Any mappings that are returned in the list for the volume are automatically stopped when the volume is deleted with the **force** option.

An invocation example

```
lsrmvdiskdependentmaps -delim : 0
```

The resulting output:

```
id:name
2:fcmap2
5:fcmap5
```

mkfcconsistgrp

Use the **mkfcconsistgrp** command to create a new FlashCopy consistency group and identification name.

Syntax

```
►►── mkfcconsistgrp ── ─┬──────────────────────────────┬─┬──────────────┬──────────────►◄
                        └─ -name ── consist_group_name ─┘ └─ -autodelete ─┘
```

Parameters

-name *consist_group_name*
(Optional) Specifies a name for the consistency group. If you do not specify a consistency group name, a name is automatically assigned to the consistency group. For example, if the next available consistency group ID is `id=2`, the consistency group name is `fccstgrp2`.

Note: Consistency group names must be an alphanumeric string of up to 15 characters.

-autodelete
(Optional) Deletes the consistency group when the last mapping that it contains is deleted or removed from the consistency group.

Description

This command creates a new consistency group and identification name. The ID of the new group is displayed when the command process completes.

If you have created several FlashCopy mappings for a group of volumes that contain elements of data for the same application, you might find it convenient to assign these mappings to a single FlashCopy consistency group. You can then issue a single prepare command and a single start command for the whole group, for example, so that all of the files for a particular database are copied at the same time.

Note: Maps that are `rc_controlled` are not shown in the view when this command is specified.

Remember: Names representing Metro Mirror or Global Mirror consistency groups relationships are restricted to fifteen characters in length (not sixty-three for an extended character set).

An invocation example

```
mkfcconsistgrp
```

The resulting output:

```
FlashCopy Consistency Group, id [1], successfully created
```

mkfcmap

Use the **mkfcmap** command to create a new FlashCopy mapping, which maps a source volume to a target volume for subsequent copying.

Syntax

```
►►── mkfcmap ── ── -source ──┬─ src_vdisk_id ───┬── ── -target ──┬─ target_vdisk_id ───┬── ──────────────►
                             └─ src_vdisk_name ─┘                 └─ target_vdisk_name ─┘

►─┬────────────────────────┬─ ─┬──────────────────────────────────────────┬─ ──────────────────►
  └─ -name ── new_name_arg ─┘   └─ -consistgrp ──┬─ consist_group_id ───┬─┘
                                                 └─ consist_group_name ─┘

►─┬────────────────────┬─ ─┬──────────────┬─┬─────────────────────┬─┬──────────────┬──────────►
  └─ -copyrate ── rate ─┘   └─ -autodelete ─┘ └─ -grainsize ──┬─ 64 ──┬─┘ └─ -incremental ─┘
                                                              └─ 256 ─┘
```

```
►─┬──────────────────────┬──┬──────────────────────────────┬──┬───────────────┬──────────────────────►◄
  └─ -cleanrate ── rate ─┘  └─ -iogrp ──┬─ iogroup_name ─┬─┘  └─ -keeptarget ─┘
                                        └─ iogroup_id ───┘
```

Parameters

-source *src_vdisk_id* **|** *src_vdisk_name*
(Required) Specifies the ID or name of the source volume.

-target *target_vdisk_id* **|** *target_vdisk_name*
(Required) Specifies the ID or name of the target volume.

-name *new_name_arg*
(Optional) Specifies the name to assign to the new mapping.

-consistgrp *consist_group_id* **|** *consist_group_name*
(Optional) Specifies the consistency group to add the new mapping to. If you do not specify a consistency group, the mapping is treated as a stand-alone mapping.

-copyrate *rate*
(Optional) Specifies the copy rate. The *rate* value can be 0 - 150. The default value is `50`. A value of `0` indicates no background copy process. For the supported **-copyrate** values and their corresponding rates, see Table 81 on page 472.

-autodelete
(Optional) Specifies that a mapping is to be deleted when the background copy completes. The default, which applies if this parameter is not entered, is that **autodelete** is set to off.

-grainsize 64 | 256
(Optional) Specifies the grain size for the mapping. The default value is 256. After set, this value cannot be changed.

Remember: If either the source or target disk contains compressed copies, the default value is 64 (unless source or target disk is part of a mapping with grain size 256 KB).

-incremental
(Optional) Marks the FlashCopy mapping as an incremental copy. The default is nonincremental. After set, this value cannot be changed.

-cleanrate *rate*
(Optional) Sets the cleaning rate for the mapping. The *rate* value can be 0 - 150. The default value is 50.

-iogrp *iogroup_name* **|** *iogroup_id*
(Optional) Specifies the I/O group for the FlashCopy bitmap. After set, this value cannot be changed. The default I/O group is either the source volume, if a single target map, or the I/O group of the other FlashCopy mapping to which either the source or target volumes belong.

Note: If not enough bitmap space is available to complete this command, more space is automatically allocated in the bitmap memory (unless the maximum bitmap memory is reached).

-keeptarget
(Optional) Specifies that the target volume and source volume availability should be kept the same. If the target becomes unavailable, the source is also made unavailable (instead of stopping the FlashCopy mapping).

Description

This command creates a new FlashCopy mapping. This mapping persists until it is manually deleted, or until it is automatically deleted when the background copy completes and the **autodelete** parameter is set to `on`. The source and target volumes must be specified on the **mkfcmap** command. The **mkfcmap** command fails if the source and target volumes are not identical in size. Issue the **lsvdisk -bytes** command to find the exact size of the source volume for which you want to create a target disk of the same size. The target volume that you specify cannot be a target volume in an existing FlashCopy mapping. A mapping cannot be created if the resulting set of connected mappings exceeds 256 connected mappings.

The mapping can optionally be given a name and assigned to a consistency group, which is a group of mappings that can be started with a single command. These are groups of mappings that can be processed at the same time. It enables multiple volumes to be copied at the same time, which creates a consistent copy of multiple disks. This consistent copy of multiple disks is required by some database products in which the database and log files reside on different disks.

If the specified source and target volumes are the target and source volumes, respectively, of an existing mapping, then the mapping that is being created and the existing mapping become partners. If one mapping is created as incremental, then its partner is automatically incremental. A mapping can have only one partner.

You can create a FlashCopy mapping in which the target volume is a member of a Metro Mirror or Global Mirror relationship, unless one of the following conditions applies:

- The relationship is with a clustered system that is running an earlier code level.
- The I/O group for the mapping is different than the I/O group for the proposed mapping target volume.

Note: You cannot use this command if a volume is part of a mapping and cloud snapshot is enabled on the volume.

The **copyrate** parameter specifies the copy rate. If `0` is specified, background copy is disabled. The **cleanrate** parameter specifies the rate for cleaning the target volume. The cleaning process is only active if the mapping is in the copying state and the background copy is completed, the mapping is in the copying state and the background copy is disabled, or the mapping is in the stopping state. You can disable cleaning when the mapping is in the copying state by setting the **cleanrate** parameter to `0`. If the **cleanrate** is set to `0`, the cleaning process runs at the default rate of 50 when the mapping is in the stopping state to ensure that the stop operation completes.

Note: You cannot issue this command if any of the following conditions apply:

- The target volume is a master or auxiliary volume in an `active-active` relationship.
- The source volume is a master or auxiliary volume in an `active-active` relationship and the target volume and map are not in the same site as the source volume.

This table provides the relationship of the copy rate and cleaning rate values to the attempted number of grains to be split per second. A grain is the unit of data that is represented by a single bit.

Note: The default grain size of 64 KB for compressed volumes applies only to compressed volumes in regular pools if:

- Either the source or target volumes are compressed within a regular pool.
- Either the source or target volumes are themselves sources or targets of compressed volumes in regular pools (i.e. in a cascade where other volumes in the cascade are compressed volumes in a regular pool).

Otherwise, the default grain size is 256 KB for data reduction compressed volumes, provided the volumes are not involved in cascades with compressed volumes in regular pools.

Table 81. Relationship between the rate, data rate, and grains per second values

User-specified *rate* **attribute value**	**Data copied/sec**	**256 KB grains/sec**	**64 KB grains/sec**
1 - 10	128 KB	0.5	2
11 - 20	256 KB	1	4
21 - 30	512 KB	2	8
31 - 40	1 MB	4	16
41 - 50	2 MB	8	32
51 - 60	4 MB	16	64
61 - 70	8 MB	32	128
71 - 80	16 MB	64	256
81 - 90	32 MB	128	512
91 - 100	64 MB	256	1024
101 - 110	128 MB	512	2048
111 - 120	256 MB	1024	4096
121 - 130	512 MB	2048	8192
131 - 140	1 GB	4096	16384
141 - 150	2 GB	8192	32768

Note: Maps that are rc_controlled are not shown in the view when this command is specified.

An invocation example

```
mkfcmap -source 0 -target 2 -name mapone
```

The resulting output:

```
FlashCopy Mapping, id [1], successfully created
```

An invocation example

```
mkfcmap -source 0 -target 2 -name mapone -keeptarget
```

The resulting output:

```
FlashCopy Mapping, id [1], successfully created
```

prestartfcconsistgrp

Use the **prestartfcconsistgrp** command to prepare a consistency group (a group of FlashCopy mappings) so that the consistency group can be started. This command flushes the cache of any data that is destined for the source volume and forces the cache into the write-through mode until the consistency group is started.

Syntax

```
►►— prestartfcconsistgrp — ————————————— ——┬— fc_consist_group_id ——┬—————————————————————►◄
                            └ -restore ┘    └ fc_consist_group_name ┘
```

Parameters

-restore
(Optional) Specifies the restore flag. This forces the consistency group to be prepared even if the target volume of one of the mappings in the consistency group is being used as a source volume of another active mapping. An active mapping is in the copying, suspended, or stopping state.

fc_consist_group_id | *fc_consist_group_name*
(Required) Specifies the name or ID of the consistency group that you want to prepare.

Description

This command prepares a consistency group (a group of FlashCopy mappings) to subsequently start. The preparation step ensures that any data that resides in the cache for the source volume is first flushed to disk. This step ensures that the FlashCopy target volume is identical to what has been acknowledged to the host operating system as having been written successfully to the source volume.

You can use the **restore** parameter to force the consistency group to be prepared even if the target volume of one or more mappings in the consistency group is being used as a source volume of another active mapping. In this case the mapping restores as shown in the **lsfcmap** view. If the **restore** parameter is specified when preparing a consistency group where none of the target volumes are the source volume of another active mapping, then the parameter is ignored.

You must issue the **prestartfcconsistgrp** command to prepare the FlashCopy consistency group before the copy process can be started. When you have assigned several mappings to a FlashCopy consistency group, you must issue a single prepare command for the whole group to prepare all of the mappings at once.

The consistency group must be in the idle_or_copied or stopped state before it can be prepared. When you enter the **prestartfcconsistgrp** command, the group enters the preparing state. After the preparation is complete, the consistency group status changes to prepared. At this point, you can start the group.

If FlashCopy mappings are assigned to a consistency group, the preparing and the subsequent starting of the mappings in the group must be performed on the consistency group rather than on an individual FlashCopy mapping that is assigned to the group. Only stand-alone mappings, which are mappings that are not assigned to a consistency group, can be prepared and started on their own. A FlashCopy consistency group must be prepared before it can be started.

This command is rejected if the target of a FlashCopy mapping in the consistency group is in a remote copy relationship, unless the relationship is one of the following types and is the secondary target of the remote copy:

- idling
- disconnected
- consistent_stopped
- inconsistent_stopped

The FlashCopy mapping also fails in the following cases:

- You use the **prep** parameter.
- The target volume is an active remote copy primary or secondary volume.
- The FlashCopy target (and remote copy primary target) volume is offline. If this occurs, the FlashCopy mapping stops and the target volume remains offline.

Note: Maps that are `rc_controlled` are not shown in the view when this command is specified.

An invocation example

```
prestartfcconsistgrp 1
```

The resulting output:

```
No feedback
```

prestartfcmap

Use the **prestartfcmap** command to prepare a FlashCopy mapping so that it can be started. This command flushes the cache of any data that is destined for the source volume and forces the cache into the write-through mode until the mapping is started.

Syntax

```
►►— prestartfcmap — ─┬────────────┬─ ─┬─ fc_map_id ───┬──────────────────────────►◄
                     └─ -restore ─┘   └─ fc_map_name ─┘
```

Parameters

-restore
(Optional) Specifies the restore flag. This forces the mapping to be prepared even if the target volume is being used as a source volume in another active mapping. An active mapping is in the copying, suspended, or stopping state.

fc_map_id | *fc_map_name*
(Required) Specifies the name or ID of the mapping to prepare.

Description

This command prepares a single mapping for subsequent starting. The preparation step ensures that any data that resides in the cache for the source volume is first transferred to disk. This step ensures that the copy that is made is consistent with what the operating system expects on the disk.

The **restore** parameter can be used to force the mapping to be prepared even if the target volume is being used as a source volume of another active mapping. In this case, the mapping is restoring as shown in the **lsfcmap** view. If the **restore** parameter is specified when preparing a mapping where the target volume is not the source volume of another active mapping, then the parameter is ignored.

Note: To prepare a FlashCopy mapping that is part of a consistency group, you must use the **prestartfcconsistgrp** command.

The mapping must be in the `idle_or_copied` or `stopped` state before it can be prepared. When the **prestartfcmap** command is processed, the mapping enters the preparing state. After the preparation is complete, it changes to the prepared state. At this point, the mapping is ready to start.

Attention: This command can take a considerable amount of time to complete. For example, while a volume is in the `prepared` state, response times might increase.

This command is rejected if the target of the FlashCopy mappings is the secondary volume in a Metro Mirror, Global Mirror, or `active-active` relationship (so that the FlashCopy target is the remote copy secondary). Remote copy includes Metro Mirror, Global Mirror, and `active-active`.

Note: If the remote copy is idling or disconnected, even if the FlashCopy and remote copy are pointing to the same volume, the auxiliary volume is not necessarily the secondary volume. In this case, you can start a FlashCopy mapping.
The FlashCopy mapping also fails in the following cases:

- The remote copy is active.
- The FlashCopy target (and remote copy primary target) volume is offline. If this occurs, the FlashCopy mapping stops and the target volume remains offline.

Note: Maps that are `rc_controlled` are not shown in the view when this command is specified.

An invocation example

```
prestartfcmap 1
```

The resulting output:

```
No feedback
```

rmfcconsistgrp

Use the **rmfcconsistgrp** command to delete a FlashCopy consistency group.

Syntax

```
►►── rmfcconsistgrp ── ┬──────────┬── ┬─ fc_consist_group_id ───┬──────────────────────────►◄
                       └─ -force ─┘   └─ fc_consist_group_name ─┘
```

Parameters

-force
(Optional) Specifies that all of the mappings that are associated with a consistency group that you want to delete are removed from the group and changed to stand-alone mappings. This parameter is only required if the consistency group that you want to delete contains mappings.

Important: Using the force parameter might result in a loss of access. Use it only under the direction of your product support information.

fc_consist_group_id | *fc_consist_group_name*
(Required) Specifies the ID or name of the consistency group that you want to delete.

Description

This command deletes the specified FlashCopy consistency group. If there are mappings that are members of the consistency group, the command fails unless you specify the **-force** parameter. When you specify the **-force** parameter, all of the mappings that are associated with the consistency group are removed from the group and changed to stand-alone mappings.

To delete a single mapping in the consistency group, you must use the **rmfcmap** command.

Note: Maps that are `rc_controlled` are not shown in the view when this command is specified.

An invocation example

```
rmfcconsistgrp fcconsistgrp1
```

The resulting output:

```
No feedback
```

rmfcmap

Use the **rmfcmap** command to delete an existing mapping.

Syntax

```
►►── rmfcmap ── ─┬────────┬─ ─┬─ fc_map_id ───┬──────────────────────────────────────►◄
                 └ -force ┘   └─ fc_map_name ─┘
```

Parameters

-force
(Optional) Specifies that the target volume is brought online. This parameter is required if the FlashCopy mapping is in the stopped state.

fc_map_id | *fc_map_name*
(Required) Specifies the ID or name of the FlashCopy mapping to delete. Enter the ID or name last on the command line.

Description

The **rmfcmap** command deletes the specified mapping if the mapping is in the `idle_or_copied` or `stopped` state. If it is in the stopped state, the **-force** parameter is required. If the mapping is in any other state, you must stop the mapping before you can delete it.

Deleting a mapping only deletes the logical relationship between the two volumes; it does not affect the volumes themselves. However, if you force the deletion, the target volume (which might contain inconsistent data) is brought back online.

If the target of the FlashCopy mapping is a member of the remote copy created , the remote copy can be affected in the following ways:

- If a stopped FlashCopy mapping is deleted and the I/O group associated with the FlashCopy mapping is suspended while this delete is being processed, then all remote copy relationships associated with the target volume of a the FlashCopy mapping that were active while the FlashCopy mapping was copying can be corrupted. You must resynchronize them next time you start the system.
- If a stopped FlashCopy mapping that has previously failed to prepare is deleted, then all remote copy relationships in the set of remote copy relationships associated with the target volume can be corrupted. You must resynchronize them next time you start the system.

Note: Remote copy includes Metro Mirror, Global Mirror, and HyperSwap.

Note: Maps that are `rc_controlled` are not shown in the view when this command is specified.

An invocation example

```
rmfcmap testmap
```

The resulting output:

```
No feedback
```

startfcconsistgrp

Use the **startfcconsistgrp** command to start a FlashCopy consistency group of mappings. This command makes a point-in-time copy of the source volumes at the moment that the command is started.

Syntax

```
►►── startfcconsistgrp ── ─┬───────┬─ ─┬──────────┬─ ─┬─ fc_consist_group_id ───┬──────────►◄
                           └ -prep ┘   └ -restore ┘   └─ fc_consist_group_name ─┘
```

Parameters

-prep
(Optional) Specifies that the designated FlashCopy consistency group be prepared prior to starting the FlashCopy consistency group. A FlashCopy consistency group must be prepared before it can be started. When you use this parameter, the system automatically issues the **prestartfcconsistgrp** command for the group that you specify.

-restore
(Optional) Specifies the restore flag. When combined with the **prep** option, this forces the consistency group to be prepared even if the target volume of one of the mappings in the consistency group is being used as a source volume in another active mapping. An active mapping is in the copying, suspended, or stopping state.

fc_consist_group_id | *fc_consist_group_name*
(Required) Specifies the ID or name of the consistency group mapping to start.

Description

This command starts a consistency group, which results in a point-in-time copy of the source volumes of all mappings in the consistency group. You can combine the **restore** parameter with the **prep** parameter to force the consistency group to be prepared prior to starting, even if the target volume of one or more mappings in the consistency group is being used as a source volume of another active mapping. In this case, the mapping is restoring as shown in the **lsfcmap** view. If the **restore** parameter is specified when starting a consistency group where none of the target volumes are the source volume of another active mapping, the parameter is ignored.

If a consistency group is started and the target volume of the mapping being started has up to four other incremental FlashCopy mappings using the target, the incremental recording is left on. If there are more than four other incremental FlashCopy mappings using the target volume, the incremental recording for all of these mappings is turned off until they are restarted.

Note: The **startfcconsistgrp** command can take some time to process particularly if you have specified the **prep** parameter. If you use the **prep** parameter, you give additional processing control to the system because the system must prepare the mapping before the mapping is started. If the prepare process takes too long, the system completes the prepare but does not start the consistency group. In this case, error message CMMVC6209E displays. To control the processing times of the **prestartfcconsistgrp** and **startfcconsistgrp** commands independently of each other, do not use the **prep** parameter. Instead, first issue the **prestartfcconsistgrp** command, and then issue the **startfcconsistgrp** command to start the copy.

This command is rejected if the target of the FlashCopy mapping in the specified consistency group is the secondary volume in a remote copy relationship (so that the FlashCopy target is the remote copy secondary).

Note: If the remote copy is idling or disconnected, even if the FlashCopy and remote copy are pointing to the same volume, the auxiliary volume is not necessarily the secondary volume. In this case, you can start a FlashCopy mapping.

The FlashCopy mapping also fails in the following cases, if the target of the FlashCopy mapping in the specified consistency group is the primary volume in a remote copy relationship (so that the FlashCopy target is the remote copy primary target:

- The remote copy is active.
- The FlashCopy target (and remote copy primary target) volume is offline. If this occurs, the FlashCopy mapping stops and the target volume remains offline.

Maps that are `rc_controlled` are not shown in the view when this command is specified.

If any source volumes in the FlashCopy consistency group are in an `active-active` relationship, the group can only be started if the information on all those source volumes current, or an older copy to which access has been provided by specifying:

```
stoprcrelationship -access
```

A current volume in an `active-active` relationship is the primary copy, or the secondary copy when the relationship's state is `consistent_syncronized`.

An invocation example

```
startfcconsistgrp -prep 2
```

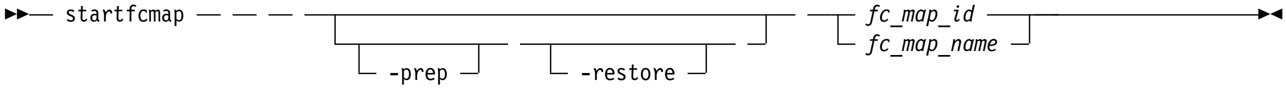
The resulting output:

```
No feedback
```

startfcmap

Use the **`startfcmap`** command to start a FlashCopy mapping. This command makes a point-in-time copy of the source volume at the moment that the command is started.

Syntax



Parameters

-prep
(Optional) Specifies that the designated mapping be prepared prior to starting the mapping. A mapping must be prepared before it can be started. When you use this parameter, the system automatically issues the **`prestartfcmap`** command for the group that you specify.

Note: If you have already used the **`prestartfcmap`** command, you cannot use the **`-prep`** parameter on the **`startfcmap`** command; the command fails. However, if the FlashCopy has successfully prepared before, the **`startfcmap`** command succeeds.

-restore
(Optional) Specifies the restore flag. When combined with the **`prep`** option, this forces the mapping to be prepared even if the target volume is being used as a source volume in another active mapping. An active mapping is in the copying, suspended, or stopping state.

fc_map_id | *fc_map_name*
Specifies the ID or name of the mapping to start.

Description

This command starts a single mapping, which results in a point-in-time copy of the source volume. You can combine the **`restore`** parameter with the **`prep`** parameter to force the mapping to be prepared prior to starting, even if the target volume is being used as a source volume of another active mapping. In this case, the mapping is restoring as shown in the **`lsfcmap`** view. If the **`restore`** parameter is specified when starting a mapping where the target volume is not the source volume of another active mapping, the parameter is ignored and the mapping is not restoring as shown in the **`lsfcmap`** view.

If a mapping is started and the target volume of the mapping being started has up to four other incremental FlashCopy mappings using the target, the incremental recording is left on. If there are more than four other incremental FlashCopy mappings using the target volume, the incremental recording for all of these mappings is turned off until they are restarted.

Note: The **startfcmap** command can take some time to start, particularly if you use the **prep** parameter. If you use the **prep** parameter, you give additional starting control to the system. The system must prepare the mapping before the mapping is started. To keep control when the mapping starts, you must issue the **prestartfcmap** command before you issue the **startfcmap** command.

This command is rejected if the target of the FlashCopy mapping is the secondary volume in a Metro Mirror or Global Mirror relationship (so that the FlashCopy target is the remote copy secondary).

Note: If the remote copy is idling or disconnected, even if the FlashCopy and remote copy are pointing to the same volume, the auxiliary volume is not necessarily the secondary volume. In this case, you can start a FlashCopy mapping.

The FlashCopy mapping also fails in the following cases, if the target of the FlashCopy mapping is the primary volume in a Metro Mirror or Global Mirror relationship (so that the FlashCopy target is the remote copy primary):

- The remote copy is active.
- The FlashCopy target (and remote copy primary target) volume is offline. If this occurs, the FlashCopy mapping stops and the target volume remains offline.

Maps that are `rc_controlled` are not shown in the view when this command is specified.

Remember: If the source volume is in an `active-active` relationship then the FlashCopy mapping can only be started if the information on the source volume is current, or an older copy to which access has been provided by specifying:

```
stoprcrelationship -access
```

A current volume in an `active-active` relationship is the primary copy, or the secondary copy when the relationship's state is `consistent_syncronized`.

An invocation example

```
startfcmap -prep 2
```

The resulting output:

```
No feedback
```

stopfcconsistgrp

Use the **stopfcconsistgrp** command to stop all processing that is associated with a FlashCopy consistency group that is in one of the following processing states: prepared, copying, stopping, or suspended.

Syntax

```
►►── stopfcconsistgrp ──┬─────────┬── fc_consist_group_id_or_name ──────────────►◄
                        ├─ -force─┤
                        └─ -split─┘
```

Parameters

-force
(Optional) Specifies that all processing that is associated with the mappings of the designated consistency group be stopped immediately.

Note: When you use this parameter, all FlashCopy mappings that depend on the mappings in this group (as listed by the **lsfcmapdependentmaps** command) are also stopped.

If the **-force** parameter is not specified, the command is rejected if the target volume of the FlashCopy consistency group is the primary in a relationship that is mirroring I/O:

- consistent_synchronized
- consistent_copying
- inconsistent_copying

If the **-force** parameter is specified, any Metro Mirror or Global Mirror relationships associated with the target volumes of the FlashCopy mappings in the specified consistency group stops. If a remote copy relationship associated with the target was mirroring I/O when the map was copying, it might lose its difference recording capability and require a full resychronization upon a subsequent restart.

-split
(Optional) Breaks the dependency on the source volumes of any mappings that are also dependent on the target volume. This parameter can only be specified when stopping a consistency group where all maps in the group have progress of 100 as shown by the **lsfcmap** command.

fc_consist_group_id_or_name
(Required) Specifies the name or ID of the consistency group that you want to stop.

Description

This command stops a group of mappings in a consistency group. If the copy process is stopped, the target disks become unusable unless they already contain complete images of the source. Disks that contain complete images of the source have a progress of 100, as indicated in the **lsfcmap** command output. The target volume is reported as offline if it does not contain a complete image. Before you can access this volume, the group of mappings must be prepared and restarted.

If the consistency group is in the idle_or_copied state, the stopfcconsistgrp command has no effect and the consistency group stays in the idle_or_copied state.

Note: Prior to SAN Volume Controller 4.2.0, the **stopfcconsistgrp** command always caused the consistency group to go to the stopped state, taking the target volumes offline.

The **split** option can be used when all of the maps in the group have progress of 100. It removes the dependency of any other maps on the source volumes. It might be used prior to starting another FlashCopy consistency group whose target disks are the source disks of the mappings being stopped. Once the consistency group has been stopped with the **split** option, the other consistency group could then be started without the **restore** option.

Note: Maps that are rc_controlled are not shown in the view when this command is specified.

An invocation example

```
stopfcconsistgrp testmapone
```

The resulting output

```
No feedback
```

stopfcmap

Use the **stopfcmap** command to stop all processing that is associated with a FlashCopy mapping that is in one of the following processing states: prepared, copying, stopping, or suspended.

Syntax

```
►►── stopfcmap ──┬───────────┬── fc_map_id_or_name ──────────────────────────────►◄
                 ├─ -force ──┤
                 └─ -split ──┘
```

Parameters

-force
(Optional) Specifies that all processing that is associated with the designated mapping be stopped immediately.

Note: When you use this parameter, all FlashCopy mappings that depend on this mapping (as listed by the **lsfcmapdependentmaps** command) are also stopped.

If the **-force** parameter is not specified, the command is rejected if the target volume of the FlashCopy mapping is the primary in a relationship which is mirroring I/O:

- consistent_synchronized
- consistent_copying
- inconsistent_copying

If the **-force** parameter is specified to a FlashCopy mapping whose target volume is also in a Metro Mirror or Global Mirror relationship, the relationship stops. If a remote copy relationship associated with the target was mirroring I/O when the map was copying, it might lose its difference recording capability and require a full resychronization on a subsequent restart.

-split
(Optional) Breaks the dependency on the source volume of any mappings that are also dependent on the target disk. This parameter can only be specified when stopping a map that has progress of 100 as shown by the **lsfcmap** command.

fc_map_id_or_name
(Required) Specifies the name or ID of the mapping to stop.

Description

This command stops a single mapping. If the copy process is stopped, the target disk becomes unusable unless it already contained a complete image of the source (that is, unless the map had a progress of 100 as shown by the **lsfcmap** command). Before you can use the target disk, the mapping must once again be prepared and then reprocessed (unless the target disk already contained a complete image).

Only stand-alone mappings can be stopped using the **stopfcmap** command. Mappings that belong to a consistency group must be stopped using the **stopfcconsistgrp** command.

If the mapping is in the `idle_or_copied` state, the **stopfcmap** command has no effect and the mapping stays in the `idle_or_copied` state.

Note: Before SAN Volume Controller 4.2.0, the **stopfcmap** command always changed the mapping state to stopped and took the target volume offline. This change can break scripts that depend on the previous behavior.

The **split** option can be used when the mapping has progress of 100. It removes the dependency of any other mappings on the source volume. It might be used prior to starting another FlashCopy mapping whose target disk is the source disk of the mapping being stopped. Once the mapping has been stopped with the **split** option, the other mapping could then be started without the **restore** option.

Note: Maps that are `rc_controlled` are not shown in the view when this command is specified.

Remember: If the source volume is in an `active-active` relationship then the FlashCopy mapping can only be stopped if the information on the source volume is current, or an older copy to which access has been provided by specifying:

```
stoprcrelationship -access
```

A current volume in an `active-active` relationship is the primary copy, or the secondary copy when the relationship's state is `consistent_syncronized`.

An invocation example

```
stopfcmap testmapone
```

The resulting output

```
No feedback
```

Chapter 17. Host commands

Use the host commands to work with host objects on your system.

addhostclustermember

Use the **addhostclustermember** command to add a host object to a host cluster.

Syntax

```
►►── addhostclustermember ── -host ──┬─ host_id_list ───┬──┬─ hostcluster_id ───┬────────────────────────►◄
                                     └─ host_name_list ─┘  └─ hostcluster_name ─┘
```

Parameters

-host *host_id_list* | *host_name_list*
(Optional) Specifies the host (by ID or name) to add to the host cluster.

hostcluster_id | *hostcluster_name*
(Required) Specifies (by ID or name) the host cluster that the host object is added to. The value for the ID must be a number and the value for the name must be an alphanumeric string.

Description

This command adds a host object to a host cluster.

When you add a host object to a host cluster, shared mappings are created. For example, if any host mappings match a host cluster mapping that is part of the same volume on the same Small Computer System Interface (SCSI) logical unit number (LUN) - with the same I/O groups - the host cluster assumes control of the mapping (which makes it a shared mapping).

Note: A host cannot be added to a host cluster if both have their individual throttling specifications defined. However, if either the host or host cluster throttling specification is present, the command succeeds.

Any mappings that do not match the shared host cluster mappings are managed by the host as private mappings.

Note: New mappings must not conflict with a shared mapping on a host system. The command fails when there are shared mappings that conflict with the host's private mappings. This includes either:

- A volume that is being mapped - but with different SCSI LUNs
- The host that has a different volume mapped but with the same SCSI LUN as a shared mapping of the host cluster

An invocation example that adds host 0 to host cluster 4

```
addhostclustermember -host 0 4
```

The resulting output:

```
No feedback
```

An invocation example that adds hosts `0`, `1`, and `4` to host cluster `4`

```
addhostclustermember -host 0:1:4 4
```

The resulting output:

```
No feedback
```

addhostiogrp

Use the **addhostiogrp** command to map I/O groups to an existing host object.

Syntax

```
►►── addhostiogrp ── ┬─ -iogrp ── iogrp_list ─┬─ ┬─ host_name ─┬──────────────────►◄
                     └─ -iogrpall ────────────┘  └─ host_id ───┘
```

Parameters

-iogrp *iogrp_list*
(Required if you do not use **-iogrpall**) Specifies a colon-separated list of one or more I/O groups that must be mapped to the host. You cannot use this parameter with the **-iogrpall** parameter.

-iogrpall
(Required if you do not use **-iogrp**) Specifies that all the I/O groups must be mapped to the specified host. You cannot use this parameter with the **-iogrp** parameter.

host_id | *host_name*
(Required) Specifies the host to which the I/O groups must be mapped, either by ID or by name.

Description

This command allows you to map the list of I/O groups to the specified host object.

An invocation example

```
addhostiogrp -iogrpall testhost
```

The resulting output:

```
No feedback
```

addhostport

Use the **addhostport** command to add worldwide port names (WWPNs) or Internet Small Computer System Interface (iSCSI) names from an existing host object.

Syntax

```
►►── addhostport ── ┬─ -saswwpn ── wwpn_list ──────┬─ ┬──────────┬─ ┬─ host_name ─┬──────────►◄
                    ├─ -fcwwpn ── wwpn_list ───────┤  └─ -force ─┘  └─ host_id ───┘
                    └─-iscsiname──iscsi_name_list──┘
```

Parameters

-saswwpn *wwpn_list*
(Required if you do not use **-iscsiname** or **-fcwwpn**) Specifies a list of Serial Attached SCSI (SAS) WWPNs with a 16-character hexadecimal string.

-fcwwpn *wwpn_list*
(Required if you do not use **-iscsiname** or **-saswwpn**) Specifies a list of Fibre Channel (FC) WWPNs with a 16-character hexadecimal string.

-iscsiname *iscsi_name_list*
(Required if you do not use **-fcwwpn** or **saswwpn**) Specifies the comma-separated list of iSCSI names to add to the host. At least one WWPN or iSCSI name must be specified. You cannot use this parameter with the **-fcwwpn** or **-saswwpn** parameter.

-force
(Optional) Specifies that the list of ports be added to the host without the validation of any WWPNs or iSCSI names.

host_id | *host_name*
(Required) Specifies the host object to add ports to, either by ID or by name.

Description

This command adds a list of host bus adapter (HBA) WWPNs or iSCSI names to the specified host object. Any volumes that are mapped to this host object automatically map to the new ports.

Only WWPNs that are logged-in unconfigured can be added. For a list of candidate WWPNs, use the **lssasportcandidate** or **lsfcportcandidate** command.

Some HBA device drivers do not log in to the fabric until they can recognize target logical unit numbers (LUNs). Because they do not log in, their WWPNs are not recognized as candidate ports. You can specify the **force** parameter with the **addhostport** command to stop the validation of the WWPN list.

Note: When all I/O groups are removed from an iSCSI host, you cannot add a port to the iSCSI host until you map the iSCSI host to at least one I/O group. After you map the iSCSI host to at least one I/O group, resubmit the **addhostport** command. After you add the port to the host, you must create a host authentication entry using the **chhost** command.

The **addhostport** command fails if the:

- Host is mapped to a volume with more than one I/O group in the access set and the host port you add is an Internet Small Computer System Interface (iSCSI) name
- Port being added is from a host system that does not support volumes mapped from multiple I/O groups

An invocation example

```
addhostport -saswwpn 210100E08B251DD4 host1
```

The resulting output:

```
No feedback
```

An invocation example

```
addhostport -fcwwpn 210100E08B251EE6 host1
```

The resulting output:

```
No feedback
```

An invocation example

```
addhostport -iscsiname iqn.localhost.hostid.7f000001 mchost13
```

The resulting output:

```
No feedback
```

chhost

Use the **chhost** command to change the name or type of a host object. This command does not affect any existing host mappings.

Syntax

```
►►── chhost ──┬─────────────────────────────────────┬─┬──────────────────────────────┬─ ──────────────►
              └─ -type ──┬─ hpux ─────────────┬───┘ └─ -mask── port_login_mask ─┘
                         ├─ tpgs──────────────┤
                         ├─ generic ──────────┤
                         ├─ openvms ──────────┤
                         ├─ adminlun ─────────┤
                         └─ hide_secondary ───┘

►─┬──────────────────────────────┬─ ─┬─────────────────────────────────────────────────────┬─ ──────────►
  └─ -name ── new_name_arg ─┘         └─ -iscsiusername ── username_for_authentication ─┘

►─┬──────────────────────────────┬─┬────────────────────────────┬─┬─ host_name ─┬────────────────────►◄
  ├─ -chapsecret ── chap_secret ─┤ ├─ -site ──┬─ site_name ─┬─┤ └─ host_id ───┘
  └─ -nochapsecret ──────────────┘ │          └─ site_id ───┘ │
                                   └─ -nosite ────────────────┘
```

Parameters

-type hpux | tpgs | generic | openvms | adminlun | hide_secondary
(Optional) Specifies the type of host. The following values are the available host types:

- `generic` indicates the default.
- `tpgs` indicates when target port information changes (extra unit attentions are given to the host).
- `openvms` indicates OpenVMS.
- `adminlun` indicates virtual volumes, which are enabled on the host.
- `hpux` indicates HP-UX firmware.
- `hide_secondary` indicates that all remote copy relationship secondary volumes are unavailable to the host.

For more information about hosts that require the **type** parameter, see the SAN Volume Controller host attachment documentation.

-name *new_name_arg*
(Optional) Specifies the new name that you want to assign to the host object.

-mask *port_login_mask*
(Optional) Specifies which node target ports a host can access and the Fibre Channel (FC) port mask for the host. Worldwide port names (WWPNs) in the host object must access volumes from the node ports that are included in the mask and are in the host object's I/O group. The port mask is 64 binary bits and is made up of a combination of 0's and 1's, where `0` indicates that the corresponding FC I/O port cannot be used and 1 indicates that it can be used. The rightmost bit in the mask corresponds to FC I/O port 1. Valid mask values might range from `0000` (no ports enabled) to 1111111111111111111111111111111111111111111111111111111111111111 (all ports enabled). For example, a mask of 111111101101 enables ports 1, 3, 4, 6, 7, 8, 9, 10, 11, and 12.

-iscsiusername *username_for_authentication*
(Optional) Specifies the user name for a host object for one-way authentication for Internet Small Computer System Interface (iSCSI) host attachment login. If this parameter is specified, the value is taken as the "username" for one-way authentication to log in to the iSCSI host attachment. If you do not specify the **iscsiusername** parameter, the IQN of the host object is used as the user name by default. If no **iscsiusername** parameter is provided and multiple IQNs exist in the same host object,

the user name of each IQN is the IQN itself. If the **iscsiusername** parameter is provided for a multiple IQN host, then log in to all hosts by using the provided IQN. If you use the **iscsiusername** parameter, you must also specify the **chapsecret** parameter.

-chapsecret *chap_secret*
(Optional) Sets the Challenge Handshake Authentication Protocol (CHAP) secret that is used to authenticate the host for iSCSI I/O. This secret is shared between the host and the cluster. The CHAP secret for each host can be listed by using the **lsiscsiauth** command.

-nochapsecret
(Optional) Clears any previously set CHAP secret for this host. The **nochapsecret** parameter cannot be specified if **chapsecret** is specified.

-site *site_name* | *site_id*
(Optional) Specifies the numeric site value or site name of the host. The site name must be an alphanumeric value. The site ID must be 1 or 2. The site that is assigned to a host can be changed with any topology: (`hyperswap`, `stretched`, or `standard`).

Note: If the host is mapped to a volume that is in an active relationship, you cannot specify **-nosite**.

-nosite
(Optional) Resets the site value.

host_name | *host_id*
(Required) Specifies the host object to modify, either by ID or by current name.

Description

This command can change the name of the specified host to a new name, or it can change the type of host. This command does not affect any of the current host mappings.

The port mask applies to logins from the host initiator port that are associated with the host object. For each login between a host bus adapter (HBA) port and node port, the node examines the port mask that is associated with the host object for which the host HBA is a member and determines whether access is allowed or is denied. If access is denied, the node responds to SCSI commands as if the HBA port is unknown.

Note: When all I/O groups are removed from an iSCSI host, the **lsiscsiauth** command does not display the authentication entry for that host. Use the **addhostiogrp** command to map the iSCSI host to at least one I/O group, and then use the **addhostport** command to add the iSCSI port into it. You must also add authentication for that host by using the **chhost** command with either the **chapsecret** or **nochapsecret** parameter.

An invocation example

```
chhost -name testhostlode -mask 111111101101 hostone
```

The following output is displayed:

```
No feedback
```

An invocation example

```
chhost -type openvms 0
```

The following output is displayed:

```
No feedback
```

An invocation example

```
chhost -site site1 host3
```

The following output is displayed:

```
No feedback
```

lshost

Use the **lshost** command to generate a list with concise information about all the hosts visible to the clustered system (system) and detailed information about a single host.

Syntax

```
▶▶── lshost── ┬─────────────────────────────────┬─ ┬────────────┬─ ┬──────────────────────────┬───▶
              └─ -filtervalue ── attribute=value ┘  └─ -nohdr ──┘   └─ -delim ── delimiter ──┘

▶─ ─ ─┬──────────────────┬─┬────────────────┬──────────────────────────────────────────────▶◀
      └─ -filtervalue? ──┘ ├─ object_id ────┤
                           └─ object_name ──┘
```

Parameters

-filtervalue *attribute=value*
: (Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are returned. If a capacity is specified, the units must also be included.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards with the SAN Volume Controller command-line interface (CLI):

- The wildcard character is an asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard character, you must enclose the filter entry within double quotation marks (""), as follows: `lshost -filtervalue "name=md*"`

-nohdr
: (Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If no data exists to be displayed, headings are not displayed.

-delim *delimiter*
: (Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (**:**) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

object_id | *object_name*
: (Optional) Specifies the name or ID of an object. When you use this parameter, the detailed view of the specific object is returned and any value that is specified by the **-filtervalue** parameter is ignored. If you do not specify the *object_id* | *object_name* parameter, the concise view of all objects that match the filtering requirements that are specified by the **-filtervalue** parameter are displayed.

-filtervalue?
(Optional) Specifies that you want your report to display any or all of the list of valid filter attributes. The valid filter attributes for the **lshost** command are:

- **host_cluster_id**
- **host_cluster_name**
- **host_name**
- **host_id**
- **id**
- **iogrp_count**
- **name**
- **port_count**
- **site_id**
- **site_name**
- **status**
- **type**

Description

This command returns a concise list or a detailed view of hosts visible to the system.

For Fibre Channel (FC) ports, the `node_logged_in_count` field provides the number of nodes that the host port is logged in to. For Internet Small Computer System Interface (iSCSI) ports, the `node_logged_in_count` field provides the number of iSCSI sessions from the host iSCSI qualified name (IQN).

You can map an iSCSI host to volumes that are accessible through multiple I/O groups. iSCSI hosts can access volumes that are accessible through multiple I/O groups (and single I/O groups). An iSCSI host that is mapped to a volume accessible through multiple I/O groups is `online` if it has at least one active iSCSI session with each I/O group of the access set. If volumes are not mapped to an iSCSI host, it is `degraded`.

The following list provides the different states for a fabric attach FC host port:

active
The host port is active if all nodes with volume mappings have a login for the specified worldwide port name (WWPN) and at least one node received SCSI commands from the WWPN within the last 5 minutes.

degraded
The host port is degraded if one or more nodes with volume mappings do not have a login for the specified WWPN.

inactive
The host port is inactive if all the nodes with volume mappings have a login for the specified WWPN but no nodes see any Small Computer System Interface (SCSI) commands from the WWPN within the last 5 minutes.

offline
The host port is offline if one or more input/output (I/O) groups with volume mappings do not have a login for the specified WWPN.

The following list provides the different states for a direct attach FC host port:

active
The host port is active if a node has a login for the specified WWPN and the node receives SCSI commands from the WWPN within the last 5 minutes.

inactive
The host port is inactive if all the nodes with volume mappings have a login for the specified WWPN but no nodes see any Small Computer System Interface (SCSI) commands from the WWPN within the last 5 minutes.

offline
The host port is offline if no login exists for the specified WWPN.

If a host does not have any volume mappings, it is reported as offline or inactive.

Note: The **`lshost`** command presents a list of host HBA ports that are logged in to nodes. However, situations exist when the information presented can include host HBA ports that are no longer logged in or even part of the SAN fabric. For example, a host HBA port is unplugged from a switch, but **`lshost`** still shows the WWPN logged in to all nodes. If this action occurs, the incorrect entry is removed when another device is plugged in to the same switch port that previously contained the removed host HBA port.

The following list provides the different states for a specified `iscsiname`:

active The `iscsiname` is active if all I/O groups with volume mappings have at least one associated iSCSI session for the specified `iscsiname`.

inactive
The `iscsiname` is inactive if the host has no volume mappings but at least one iSCSI session for the specified `iscsiname` is present.

offline
The `iscsiname` is offline if one or more I/O groups with volume mappings do not have an associated iSCSI session for the specified `iscsiname`.

The following list provides the different states for host_status:

online The host has full connectivity. A host that uses just one style of connectivity is online if it uses one of these types:

Fibre Attach Fibre Channel (FAFC)
Every port is active or inactive, and is logged in to every online node in each I/O group in which the host has volume mappings.

Direct Attach Fibre Channel (DAFC)
The host has an active or inactive login to every node in I/O groups to which the host has volume mappings.

Internet Small Computer System Interface (iSCSI)
The host has an iSCSI session with each I/O group with which the host has volume mappings.

offline
The host has no connectivity. The reason might be because the host is powered off and is not on.

Remember: If an iSCSI host is only logged in to I/O groups for which it is not configured, the associated host object status is `offline`.

degraded
The host is not fully connected, which might be introduced by a configuration error or a hardware failure. It can cause a loss of access during any planned maintenance activity and must be corrected as soon as possible.

Remember: An iSCSI host that has no mapped volumes is `degraded` if it is logged in to some, but not all, of the I/O groups to which it belongs.

mask The Fibre Channel (FC) I/O ports (which exist on a node) hosts can access.

Table 82 shows the possible outputs:

Table 82. lshost output

Attribute	Description
id	Indicates the unique host ID. The value is an alphanumeric value.
name	Indicates the unique host name. The value is an alphanumeric string.
port_count	Indicates the number of ports.
type	Indicates the host type.
mask	Indicates the mask value with a 64-bit binary string.
iogrp_count	Indicates the number of I/O groups.
status	Indicates whether the host is `online` or `offline`.
WWPN	Indicates the worldwide port name (WWPN) with a 16-character hexadecimal string.
SAS_WWPN	Indicates the serial-attached SCSI (SAS) WWPN with a 16-character hexadecimal string.
node_logged_in_count	Indicates the number of nodes the WWPN is logged in to.
state	Indicates the state of the SAS WWPN login. The values are: • `offline` • `inactive` • `active`
sas_wwpn_count	Indicates the number of configured SAS WWPNs.
site_id	Identifies the site ID for the host. The values are `1`, `2`, or blank.
site_name	Identifies the site name for the host. The value must be an alphanumeric string or blank.
host_cluster_id	Indicates the unique ID for a host cluster.
host_cluster_name	Indicates the unique name for a host cluster.

An invocation example

```
lshost
```

The resulting output:

```
id name    port_count iogrp_count status mapping_count host_cluster_id  host_cluster_name site_id site_name
  0  hostone 1          4           offline             0                vardyhost1 2 chelsea3
  1  host0   1          4           degraded            1                vardyhost2 1 chelsea1
  2  host1   1          4           online              2                vardyhost3 2 chelsea2
```

A detailed invocation example

```
lshost 0
```

The resulting output:

```
id 0
name ined
port_count 1
type openvms
mask 0000000000000000000000000000000000000000000000000000000000001101
iogrp_count 4
status online
WWPN 10000000C92BB490
node_logged_in_count 1
state inactive
```

```
site_id 2
site_name chelsea2
host_cluster_id 1
host_cluster_name jvardy8
```

An invocation example

```
lshost 0
```

The resulting output:

```
id 0
name host0
port_count 10
type generic
mask 1111111111111111111111111111111111111111111111111111111111111111
iogrp_count 4
status offline
SAS_WWPN 1000000000000009
node_logged_in_count 0
state offline
SAS_WWPN 1000000000000008
node_logged_in_count 0
state offline
site_id 2
site_name chelsea2
host_cluster_id 2
host_cluster_name boat3ng
```

lshostcluster

Use the **lshostcluster** command to generate a list with concise information about all the host clusters visible to the clustered system (system) or detailed information about a single host cluster.

Syntax

```
►►── lshostcluster ──┬─────────┬──┬──────────────────────┬──────────────────────►
                     └─ -nohdr ─┘  └─ -delim ── delimiter ─┘

►──┬──────────────────────────────────────────────────┬──────────────────────►◄
   └─ -hostcluster ──┬─ hostcluster_id ───┬─┘
                     └─ hostcluster_name ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim : on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-hostcluster *hostcluster_id* | *hostcluster_name*

(Required) Specifies the ID or name of the host to display information about. If you do not enter a host system ID or name, the command displays a list of all recognized host clusters and volume mappings. The value for the ID must be a number and the value for the name must be an alphanumeric string.

Description

This command lists concise information about all the host clusters visible to the clustered system or detailed information about a single host cluster.

This table provides the attribute values that can be displayed as output view data.

Table 83. lshostcluster output

Attribute	Description
id	Indicates the host cluster ID.
name	Indicates host cluster name. The value must be an alphanumeric string of no more than 64 characters.
status	Indicates the status of the host cluster. The values are: • `online` indicates that all hosts or members are online. • `host_degraded` indicates that no hosts are offline but at least one host is degraded. • `host_cluster_degraded` indicates that one or more hosts are offline and at least one host is online or degraded. • `offline` indicates that all hosts are offline or there are no hosts or members in the host cluster.
host_count	Indicates the number of hosts that are in the host cluster. The value must be a number in the range 0 - 127.
mapping_count	Indicates the number of shared mappings between the host cluster and any existing volumes. The value must be a number in the range 0 - 2047.
port_count	Indicates the number of host ports that are used for the host cluster mappings to any volumes. The value must be a number in the range 0 - 255.

A concise invocation example

```
lshostcluster
```

The detailed resulting output:

```
id name         status host_count mapping_count port_count
0  hostcluster0 online 2          1             4
```

A detailed invocation example

```
lshostcluster : hostcluster0
```

The detailed resulting output:

```
id:0
name:hostcluster0
status:online
host_count:6
mapping_count:32
port_count:12
```

lshostclustermember

Use the **lshostclustermember** command to generate a list with host information for hosts that belong to the specified host cluster.

Syntax

```
►►─ lshostclustermember ─┬─────────┬─┬────────────────────┬─┬─ hostcluster_id ───┬──────────►◄
                         └─ -nohdr ─┘ └─ -delim ─ delimiter ─┘ └─ hostcluster_name ─┘
```

Parameters

-nohdr

(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If no data exists to be displayed, headings are not displayed.

-delim *delimiter*

(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim : on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

hostcluster_id | *hostcluster_name*

(Required) Specifies the ID or name of the host that is part of the host cluster. If you do not enter a host cluster ID or name, the command displays a list of all recognized host clusters and volume mappings. The value for the ID must be a number and the value for the name must be an alphanumeric string.

Description

This command information about all hosts that belong to the specified host cluster.

This table provides the attribute values that can be displayed as output view data.

Table 84. **lshostclustermember** *output*

Attribute	Description
host_id	Indicates the unique ID of the host cluster. The value must be a number 0 - 4095.
host_name	Indicates the host name. The value must be an alphanumeric string of no more than 64 characters.
status	Indicates the status of a host for a host cluster. The values are: • online indicates that all hosts or members are online. • host_degraded indicates that no hosts are offline but at least one host is degraded. • offline indicates that all hosts are offline or there are no hosts or members in the host cluster.

Table 84. **lshostclustermember** *output (continued)*

Attribute	Description
type	Indicates the unique ID for the site that the host cluster is in. The values are: • `generic` • `hpux` • `tpgs` • `openmvs` • `adminlun` • `hide_secondary`
site_id	Indicates the site ID (that the host cluster is part of). The value must be a number in the range 0 - 3.
site_name	Indicates the site name (that the host cluster is part of). The value must be an alphanumeric string.

A concise invocation example

```
lshostclustermember
```

The detailed resulting output:

```
 host_id host_name status type    site_id site_name
 0       host0     online generic 1       site1
```

A detailed invocation example

```
lshostclustermember :
```

The detailed resulting output:

```
host_id:0
host_name:j1mvardy
status:online
type:generic
site_id:1
site_name:jamiev12
```

lshostclustervolumemap

Use the **lshostclustervolumemap** command to display a list of volumes that are mapped to all host clusters (or to a specific host cluster).

Syntax

```
►►── lshostclustervolumemap ──┬─────────┬──┬────────────────────────┬── ──┬─ hostcluster_id ───┬───────►◄
                              └─ -nohdr ─┘  └─ -delim ── delimiter ─┘     └─ hostcluster_name ─┘
```

Parameters

-nohdr
: (Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

  Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
: (Optional) By default in a concise view, all columns of data are space-separated. The width of each

column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (**:**) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

hostcluster_id | *hostcluster_name*
(Required) Specifies the ID or name for the host cluster that is being mapped to a volume. The command displays a list of all the volumes that are mapped to the specified host cluster and additionally indicates the Small Computer System Interface (SCSI) ID through which they are mapped. If you do not enter a host cluster ID or name, the command displays a list of all recognized host clusters and volume mappings. The value for the ID must be a number and the value for the name must be an alphanumeric string.

Description

This command displays a list of volumes that are mapped to all host clusters or to a specific host cluster.

This table provides the attribute values that can be displayed as output view data.

Table 85. lshostclustervolumemap output

Attribute	Description
id	Indicates the host cluster ID. The value must be a number in the range 0 - 127.
name	Indicates host cluster name. The value must be an alphanumeric string of no more than 64 characters.
SCSI_id	Indicates the unique ID (volume ID) that is mapped from a host cluster to an I/O group volume. The value must be a number in the range 0 - 2047.
volume_id	Indicates the unique ID of the volume that is mapped to the host cluster. The value must be a number.
volume_name	Indicates the name for a volume that is mapped to a host cluster. The value must be an alphanumeric string.
volume_UID	Indicates the unique UID of a volume. The value must be an alphanumeric string.
IO_group_id	Indicates the unique ID from the I/O group that the host cluster and volume (from the mapping) are part if. The value must be a number in the range 0 - 3.
IO_group_name	Indicates the I/O group name. The value must be an alphanumeric string.

A concise invocation example

```
lshostclustervolumemap
```

The detailed resulting output:

```
name         SCSI_id volume_id volume_name volume_UID                       IO_group_id IO_group_name
 0  hostcluster0 0       0         vdisk0      60050764009900082000000000000000 0           io_grp0
0  hostcluster0 1       1         vdisk1      60050764009900082000000000000001 0           io_grp0
0  hostcluster0 2       2         vdisk2      60050764009900082000000000000002 0           io_grp0
0  hostcluster0 3       3         vdisk3      60050764009900082000000000000003 0           io_grp0
1  hostcluster1 0       4         vdisk4      60050764009900082000000000000004 0           io_grp0
1  hostcluster1 1       5         vdisk5      60050764009900082000000000000005 0           io_grp0
```

A concise invocation example

```
lshostclustervolumemap 0
```

The detailed resulting output:

```
 id name         SCSI_id volume_id volume_name volume_UID                       IO_group_id IO_group_name
0  hostcluster0 0       0         vdisk0      60050764009900082000000000000000 0           io_grp0
0  hostcluster0 1       1         vdisk1      60050764009900082000000000000001 0           io_grp0
0  hostcluster0 2       2         vdisk2      60050764009900082000000000000002 0           io_grp0
0  hostcluster0 3       3         vdisk3      60050764009900082000000000000003 0           io_grp0
```

A concise invocation example

```
lshostclustervolumemap hostcluster1
```

The detailed resulting output:

```
id id name         SCSI_id volume_id volume_name volume_UID                       IO_group_id IO_group_name
1  hostcluster1 0       4         vdisk4      60050764009900082000000000000004 0           io_grp0
1  hostcluster1 1       5         vdisk5      60050764009900082000000000000005 0           io_grp0
```

lshostiogrp

Use the **lshostiogrp** command to display a list the I/O groups that are associated with a specified host.

Syntax

```
►►— lshostiogrp — ┬──────────┬─┬─────────────────────┬─┬─ host_id ───┬──────────────────────►◄
                  └─ -nohdr ─┘ └─ -delim — delimiter ─┘ └─ host_name ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim : on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

host_id | *host_name*
(Required) The name or ID of the host for which the list of I/O groups is required.

Description

This command displays a list of all the I/O groups that are mapped to the specified host.

An invocation example

```
lshostiogrp -delim : hostone
```

The resulting output:

```
id:name
0:io_grp0
1:io_grp1
```

lsiscsiauth

Use the **lsiscsiauth** command to list the Challenge Handshake Authentication Protocol (CHAP) secret that is configured for authenticating an entity to the system.

Syntax

```
►►— lsiscsiauth — ┬─────────┬─ — ┬────────────────────────┬──────────────────────────────►
                  └─ -nohdr ┘    └─ -delim — delimiter ─┘

►┬──────────────────────────────────┬─ ┬─────────────────┬─ ──────────────────────────►◄
 └─ -filtervalue — attribute=value ─┘  └─ -filtervalue? ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim **:** on the command line, the colon character (**:**) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed. If a capacity is specified, the units must also be included.

Note: Some filters allow the asterisk character (*) when you enter the command. The following rules apply to the use of wildcard characters with the system CLI:

- The wildcard character is an asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, you must enclose the filter entry within double quotation marks (""), as follows:

```
lsiscsiauth -filtervalue  "name=md*"
```

-filtervalue?
(Optional) displays a list of filters that can be applied against this view. The following filter attributes are valid for the **lsiscsiauth** command:

- `type`
- `id`
- `name`
- `iscsi_auth_method`
- `iscsi_chap_secret`
- `cluster_iscsi_auth_method`
- `cluster_iscsi_chap_secret`
- `iscsiusername`

Description

This command lists the CHAP secret that is configured for authenticating an entity to the system. The command also displays the configured iSCSI authentication method. The iscsi_auth_method field can have values of `none` or `chap`.

When you create an iSCSI host by using the **mkhost** command with the **iscsiname** parameter, the host is initially configured with the authentication method as `none`, and no CHAP secret is set. To set a CHAP secret for authenticating the iSCSI host with the system, use the **chhost** command with the **chapsecret** parameter.

This table provides the attribute values that can be displayed as output view data.

Table 86. lsiscsiauth output

Attribute	Description
type	Indicates the iSCSI system type.
id	Indicates the iSCSI system ID.
name	Indicates the iSCSI system name.
iscsi_auth_method	Indicates the iSCSI authentication method.
iscsi_chap_secret	Indicates whether an iSCSI CHAP secret exists.
cluster_iscsi_auth_method	Indicates clustered system iSCSI authentication method.
cluster_iscsi_chap_secret	Indicates the clustered system iSCSI configured CHAP secret.
iscsiusername	Indicates the iSCSI user name.

An invocation example

```
lsiscsiauth
```

The following output is displayed:

```
type  id  name       iscsi_auth_method  iscsi_chap_secret  cluster_iscsi_auth_method  cluster_iscsi_chap_secret
host  0   mchost20   none                                  none
host  1   mchost30   none                                  none
host  2   mchost200  none                                  none
host  3   mchost40   none                                  none
host  4   mchost240  none                                  none
host  5   mchost170  none                                  none
host  6   mchost120  none                                  none
host  7   mchost60   none                                  none
host  8   mchost180  none                                  none
host  9   mchost13   none                                  none
host  10  newhost    none                                  none
```

An invocation example

```
lsiscsiauth -iscsiusername
```

The following output is displayed:

```
type  id  name       iscsi_auth_method  iscsiusername  iscsi_chap_secret
host  0   host0      chap               rhel_host1     rhel_secret
```

An invocation example

```
iscsiusername
```

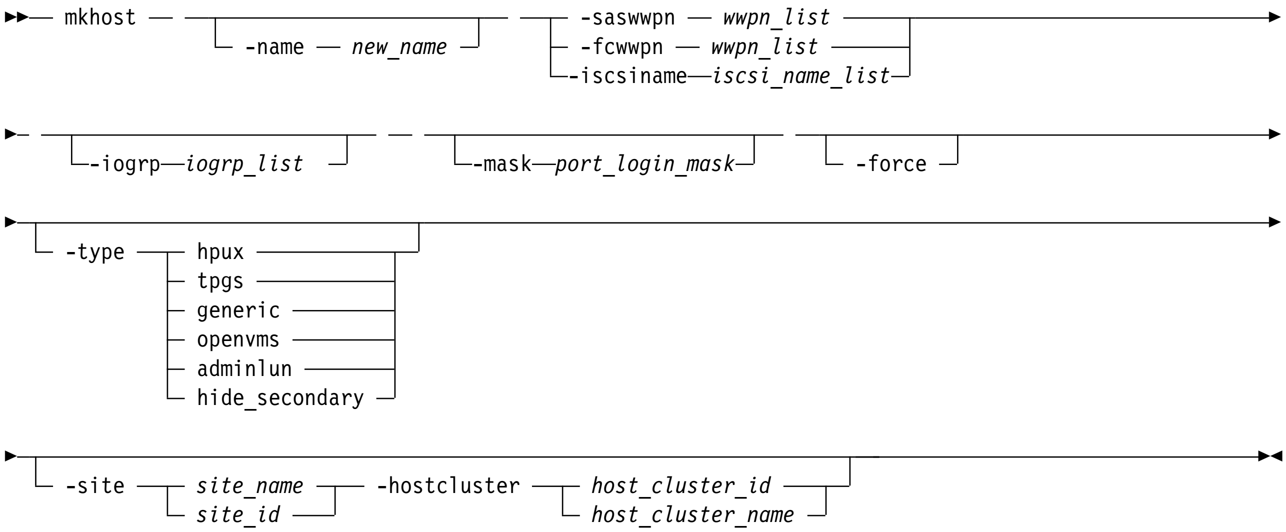
The following output is displayed:

```
type  id  name       iscsi_auth_method  iscsiusername  iscsi_chap_secret
host  0   host0      chap               -              rhel_secret
```

mkhost

Use the **mkhost** command to create a logical host object.

Syntax



```
>>— mkhost —+----------------------+— —+— -saswwpn — wwpn_list ——+——————————————>
            '— -name — new_name —'     +— -fcwwpn — wwpn_list ———+
                                       '—-iscsiname—iscsi_name_list—'

>— +————————————————————+ — +——————————————————————+ +—————————+——————————>
   '—-iogrp—iogrp_list —'   '—-mask—port_login_mask—' '— -force —'

>—+————————————————————————————+————————————————————————————————————————>
  '— -type —+— hpux ———————————+
            +— tpgs ———————————+
            +— generic ————————+
            +— openvms ————————+
            +— adminlun ———————+
            '— hide_secondary —'

>—+—————————————————————————————————————————————————————————————+————————><
  '— -site —+— site_name —+— -hostcluster —+— host_cluster_id ———+
            '— site_id ———'                '— host_cluster_name —'
```

Parameters

-name *new_name*
(Optional) Specifies a name or label for the new host object.

-saswwpn *wwpn_list*
(Required if you do not use **-iscsiname** or **-fcwwpn**) Specifies a list of Serial Attached SCSI (SAS) WWPNs with a 16-character hexadecimal string.

-fcwwpn *wwpn_list*
(Required if you do not use **-saswwpn** or **-iscsiname**) Specifies a list of Fibre Channel (FC) WWPNs with a 16-character hexadecimal string.

-iscsiname *iscsi_name_list*
(Required if you do not use **-fcwwpn** or **-saswwpn**) Specifies the comma-separated list of iSCSI names to add to the host. At least one WWPN or iSCSI name must be specified. You cannot use this parameter with the **-fcwwpn** or **-saswwpn** parameter.

-iogrp *iogrp_list*
(Optional) Specifies a set of one or more input/output (I/O) groups that the host can access the volumes from. I/O groups are specified by using their names or IDs, separated by a colon. Names and IDs can be mixed in the list. If this parameter is not specified, the host is associated with all I/O groups.

-mask *port_login_mask*
(Optional) Specifies which node target ports a host can access and the Fibre Channel (FC) port mask for the host. Worldwide port names (WWPNs) in the host object must access volumes from the node ports that are included in the mask and are in the host object's I/O group. The port mask is 64 binary bits and is made up of a combination of 0's and 1's, where `0` indicates not to use the corresponding FC I/O port and 1 indicates to use the corresponding FC I/O port. The right-most bit in the mask corresponds to FC I/O port 1. Valid mask values might range from `0000` (no ports enabled) to `1111111111111111111111111111111111111111111111111111111111111111` (all ports enabled). For example, a mask of `111111101101` enables ports 1, 3, 4, 6, 7, 8, 9, 10, 11, and 12.

-force
(Optional) Specifies that a logical host object is created without validation of the WWPNs.

-type hpux | tpgs | generic | openvms | adminlun | hide_secondary
(Optional) Specifies the type of host. The default is **generic**. The **adminlun** host type is equivalent to the **VVOL** host type in the management GUI. The **tpgs** host type enables extra target-port unit attentions and is required for any Solaris host.

-hostcluster *host_cluster_id* | *host_cluster_name*
(Optional) Specifies the host cluster ID (numerical value) or name (alphanumeric value) that the new host object is created in.

Description

The **mkhost** command associates one or more HBA WWPNs or iSCSI names with a logical host object. This command creates a new host. The ID is displayed when the command completes. Subsequently, you can use this object when you map volumes to hosts by using the **mkvdiskhostmap** command. If you create a host directly inside a host cluster by specifying **-hostcluster**, it inherits any shared mappings that exist.

Issue the **mkhost** command only once. The clustered system scans the fabric for WWPNs in the host zone. The system itself cannot filter into the hosts to determine which WWPNs are in which hosts. Therefore, you must use the **mkhost** command to identify the hosts.

After you identify the hosts, mappings are created between hosts and volumes. These mappings effectively present the volumes to the hosts to which they are mapped. All WWPNs in the host object are mapped to the volumes.

Some HBA device drivers are not logged in to the fabric until they recognize target logical unit numbers (LUNs). Because they do not log in, their WWPNs are not recognized as candidate ports. You can specify the **force** parameter with this command to stop the validation of the WWPN list.

This command fails if you add the host to an I/O group that is associated with more host ports or host objects than is allowed by the limits within the system.

For additional information, see the **mkvdiskhostmap**, **lssasportcandidate**, and **lsfcportcandidate** commands. For more information about parameter requirements for your specific host, refer to the following support site:

http://www-03.ibm.com/systems/support/storage/ssic/interoperability.wss

An invocation example

```
mkhost -name hostone -saswwpn 210100E08B251DD4:210100F08C262DD8 -force -mask 111111101101
```

The resulting output:

```
Host id [1] successfully created
```

An invocation example

```
mkhost -iscsiname iqn.localhost.hostid.7f000001 -name newhost
```

The resulting output:

```
Host, id [10], successfully created
```

An invocation example

```
mkhost -fcwwpn 210100E08B251EE6:210100F08C262EE7 -type openvms
```

The resulting output:

```
Host, id [1], successfully created
```

An invocation example

```
mkhost -fcwwpn 210100E08B251EE6 -site site1
```

The resulting output:

```
Host, id [1], successfully created
```

mkhostcluster

Use the **mkhostcluster** command to create a host cluster object.

Syntax

```
►►── mkhostcluster ── -name ── name ──────────────────────────────────────────►

►─ -seedfromhost ──┬─ host_id_list ───┬── -ignoreseedvolume ──┬─ volume_id_list ───┬──────────────►◄
                   └─ host_name_list ─┘                       └─ volume_name_list ─┘
```

Parameters

-name *name*
(Optional) Specifies the name of the host cluster object.

-seedfromhost *host_id_list* | *host_name_list*
(Optional) Adds the specified host to the host cluster. The host cluster mappings to existing volumes then become shared host cluster mappings. If a list of hosts is provided, the hosts are mapped to the same volume that uses the same I/O group and with the same SCSI LUN.

Note: These become shared mappings unless explicitly excluded with **-ignoreseedvolume**.

-ignoreseedvolume *volume_id_list* | *volume_name_list*
(Optional) Specifies volumes that are not part of the shared host cluster mappings. These volumes remain privately mapped to the host or hosts. You must specify **-seedfromhost** if you specify this parameter.

Description

This command is used to create a host cluster object.

Note: This command fails if any of the specified seeding hosts has an associated host throttle.

An invocation example that creates host cluster myhostcluster and obtains its mappings from host myhost1

```
mkhostcluster -name myhostcluster -seedfromhost myhost1
```

The detailed resulting output:

```
No feedback
```

An invocation example that creates host cluster `myhostcluster` that obtains its mappings from host `myhost1`

The system keeps the mapping to its boot drive (`volume_4`) private.

```
mkhostcluster -name myhostcluster -seedfromhost myhost1 -ignoreseedvolume volume_4
```

The detailed resulting output:

```
No feedback
```

An invocation example of a list being created

```
mkhostcluster -seedfromhost 1:2:3
```

The detailed resulting output:

```
No feedback
```

mkvolumehostclustermap

Use the **mkvolumehostclustermap** command to generate a new mapping between a volume and a host cluster on a clustered system. This volume is then accessible for input or output (I/O) operations to the specified host cluster.

Syntax

```
►►── mkvolumehostclustermap ──┬──────────────────────┬──┬──────────┬──────────────────►
                              └─ -scsi scsi_num_arg ─┘  └─ -force ─┘

►─ -hostcluster ──┬─ hostcluster_id ───┬──┬─ volume_id ───┬────────────────────────►◄
                  └─ hostcluster_name ─┘  └─ volume_name ─┘
```

Parameters

-scsi *scsi_num_arg*
(Optional) Specifies the Small Computer System Interface (SCSI) logical unit number (LUN) ID to assign to a volume on the specified host cluster. The SCSI LUN ID is assigned to the volume on the host cluster for all I/O groups that provide access to the volume.

Note: You must use the next available SCSI LUN ID for each host in the host cluster.

-force
(Optional) Forces a new mapping. Specify this parameter to map a volume to a host cluster and that volume is already mapped to at least one host in a different host cluster.

Remember: Using the force parameter might result in a loss of access. Use it only under the direction of your product support information.

-hostcluster *hostcluster_id* | *hostcluster_name*
(Required) Specifies the host cluster (by ID or name) to map to the volume. The value for the ID must be a number and the value for the name must be an alphanumeric string.

volume_id | *volume_name*
(Optional) Specifies the volume by ID or name. The value for the ID must be a number and the value for the name must be an alphanumeric string.

Description

This command generates a new mapping between a volume and a host cluster on a clustered system (system). This volume is then accessible for input or output (I/O) operations to the specified host cluster.

An invocation example that maps volume `0` to host cluster `0`

```
 mkvolumehostclustermap -hostcluster 0 0
```

The detailed resulting output:

```
No feedback
```

An invocation example that maps volume `myvolume1` to host cluster `myhostcluster` and specifies SCSI LUN ID `7`

```
mkvolumehostclustermap -hostcluster myhostcluster -scsi 7 myvolume1
```

The detailed resulting output:

```
No feedback
```

rmhost

Use the **`rmhost`** command to delete a host object.

Syntax

```
►►── rmhost ── ──┬──────────┬── ──┬─ host_name ─┬─────────────────────────────────►◄
                 └─ -force ─┘     └─ host_id ───┘
```

Parameters

`-force`
(Optional) Specifies that you want the system to delete the host object even if mappings still exist between this host and volumes. When the **`-force`** parameter is specified, the mappings are deleted before the host object is deleted.

`host_name` | *`host_id`*
(Required) Specifies the host object to delete, either by ID or by name.

Description

The **`rmhost`** command deletes the logical host object. The WWPNs that were contained by this host object (if it is still connected and logged in to the fabric) are returned to the unconfigured state. When you issue the **`lsfcportcandidate`** or **`lssasportcandidate`** command, the host objects are listed as candidate ports.

Note: This command deletes the associated host throttle if that host is removed.

Remember: This command is unsuccessful if:

- Volume protection is enabled (using the **`chsystem`** command)
- The host being deleted is mapped to any volume that has received I/O within the defined volume protection time period

If any mappings still exist between this host and volumes, the command fails unless you specify the **`-force`** parameter. When the **`-force`** parameter is specified, the **`rmhost`** command deletes the mappings before the host object is deleted.

An invocation example

```
rmhost host_one
```

The resulting output:

```
No feedback
```

rmhostcluster

Use the **rmhostcluster** command to remove a host cluster.

Syntax

```
►►── rmhostcluster ─┬──────────────────┬─┬─ hostcluster_id ───┬──────────────────────────────►◄
                    ├─ -removeallhosts ─┤ └─ hostcluster_name ─┘
                    ├─ -removemappings ─┤
                    └─ -keepmappings ───┘
```

Parameters

-removeallhosts
(Optional) Specifies the deletion of all hosts and the associated host cluster object.

-removemappings
(Optional) Specifies that the host cluster object being removed from the host cluster not use the host cluster's shared volume mappings. The mappings are deleted before the host cluster is deleted.

-keepmappings
(Optional) Specifies that the host cluster object removed from the host cluster retains the host cluster shared volume mappings (which become private mappings).

hostcluster_id | *hostcluster_name*
(Required) Specifies (by ID or name) the host cluster that the host cluster object is removed from. The value for the ID must be a number and the value for the name must be an alphanumeric string.

Description

This command removes a host cluster.

Note: This command deletes the associated host cluster throttle if that host cluster is removed.

The **-removeallhosts**, **-keepmappings**, and **-removemappings** parameters are mutually exclusive.

An invocation example that removes host cluster hostcluster0 and any related hosts

```
rmhostcluster -removeallhosts hostcluster0
```

The detailed resulting output:

```
No feedback
```

An invocation example that removes host cluster hostcluster0 and all mappings to volumes

```
rmhostcluster -removemappings hostcluster0
```

The detailed resulting output:

```
No feedback
```

An invocation example that removes host cluster `hostcluster0`

The hosts that are removed keep the shared mappings from the host cluster as private mappings.

```
rmhostcluster -keepmappings hostcluster0
```

The detailed resulting output:

```
No feedback
```

rmhostclustermember

Use the **`rmhostclustermember`** command to remove a host from a host cluster object.

Syntax

```
►►── rmhostclustermember ── -host ──┬─ host_id_list ───┬──┬──────────────────┬── -force ──────────►
                                    └─ host_name_list ─┘  ├─ -keepmappings ───┤
                                                          └─ -removemappings ─┘

►──┬─ hostcluster_id ───┬──────────────────────────────────────────────────────────────────────►◄
   └─ hostcluster_name ─┘
```

Parameters

-host *host_id_list* | *host_name_list*
(Optional) Specifies (by ID or name) the hosts to remove from the host cluster.

-keepmappings
(Optional) Specifies that the host that is removed from the host cluster retains the host cluster's shared volume mappings. The **-keepmappings** and **-removemappings** parameters are mutually exclusive.

-removemappings
(Optional) Specifies that the host that is removed from the host cluster does not retain the host cluster's shared volume mappings. The **-keepmappings** and **-removemappings** parameters are mutually exclusive.

-force
(Optional) Forces a removal. Specify this parameter when you remove the last host from a host cluster.

Remember: Using the force parameter might result in a loss of access. Use it only under the direction of your product support information.

hostcluster_id | *hostcluster_name*
(Required) Specifies (by ID or name) that the host cluster that the host is removed from. The value for the ID must be a number and the value for the name must be an alphanumeric string.

Description

This command removes a host from a host cluster object.

An invocation example that removes host `0` from host cluster `0` (and also removes the host mappings)

```
 rmhostclustermember -host 0 -removemappings 0
```

The detailed resulting output:

```
No feedback
```

An invocation example that removes host `myhost1` from host cluster `myhostcluster` while keeping the original mappings

```
rmhostclustermember -host myhost1 -keepmappings -force myhostcluster
```

The detailed resulting output:

```
No feedback
```

rmvolumehostclustermap

Use the **rmvolumehostclustermap** command to remove an existing host cluster mapping to a volume on a clustered system.

Syntax

```
►►── rmvolumehostclustermap ── ── -hostcluster ──┬─ hostcluster_id ───┬──────────────────────────────►
                                                  └─ hostcluster_name ─┘

►──┬──────────────────────────────────────┬──┬─ volume_id ───┬──────────────────────────────────►◄
   └─ -makeprivate ──┬─ host_id_list ───┬─┘  └─ volume_name ─┘
                     └─ host_name_list ─┘
```

Parameters

-hostcluster *hostcluster_id* | *hostcluster_name*
(Required) Specifies the host cluster (by ID or name) to remove from the volume mapping. The value for the ID must be a number and the value for the name must be an alphanumeric string.

-makeprivate *host_id_list* | *host_name_list*
(Optional) Specifies the host or hosts that acquire private mappings from the volume that is being removed from the host cluster. The value for the ID must be a number and the value for the name must be an alphanumeric string.

volume_id | *volume_name*
(Required) Specifies the volume by ID or name. The value for the ID must be a number and the value for the name must be an alphanumeric string.

Description

This command removes an existing host cluster mapping on a host cluster. The volume is then inaccessible for input or output (I/O) transactions from the specified host cluster.

An invocation example that moves a mapping from host cluster `0` to volume `0`

```
rmvolumehostclustermap -hostcluster 0 0
```

The resulting output:

```
No feedback
```

A detailed invocation example that removes a mapping from host cluster `myhostcluster` and adds it to volume `myvolume1`

```
rmvolumehostclustermap -hostcluster myhostcluster myvolume1
```

The resulting output:

```
No feedback
```

A detailed invocation example that removes a mapping from host cluster `myhostcluster` and adds it to volume `myvolume1`

This example allows hosts `myhost1` and `myhost2` to acquire the private mappings from `myvolume1`.

```
rmvolumehostclustermap -hostcluster myhostcluster -makeprivate myhost1:myhost2 myvolume1
```

The resulting output:

```
No feedback
```

rmhostiogrp

Use the **`rmhostiogrp`** command to delete mappings between one or more input/output (I/O) groups and a specified host object.

Syntax

```
►►── rmhostiogrp ── ─┬─ -iogrp ── ─ iogrp_list ─┬─┬─────────┬─ ─┬─ host_name ─┬──────────────►◄
                     └─ -iogrpall ──────────────┘ └─ -force ─┘  └─ host_id ───┘
```

Parameters

-iogrp *iogrp_list*
(Required) Specifies a set of one or more I/O group mappings that will be deleted from the host. You cannot use this parameter with the **iogrpall** parameter.

-iogrpall
(Optional) Specifies that all the I/O group mappings that are associated with the specified host must be deleted from the host. You cannot use this parameter with the **iogrp** parameter.

-force
(Optional) Specifies that you want the system to remove the specified I/O group mappings on the host even if the removal of a host to I/O group mapping results in the loss of host mappings.

Remember: Using the force parameter might result in a loss of access. Use it only under the direction of your product support information.

host_id | *host_name*
(Required) Specifies the identity of the host either by ID or name from which the I/O group mappings must be deleted.

Description

The **`rmhostiogrp`** command deletes the mappings between the list of I/O groups and the specified host object.

Remember: This command is unsuccessful if:

- Volume protection is enabled (using the **`chsystem`** command)
- The host I/O group being removed is mapped to any volume that has received I/O within the defined volume protection time period

If a host is defined in two I/O groups, and has access to a volume through both I/O groups, an attempt to remove the host from just one of those I/O groups fails, even with **-force** specified. To resolve this problem, do one of the following:

- Delete the host mappings that are causing the error
- Delete the volumes or the host

Note: When all I/O groups are removed from an Internet Small Computer System Interface (iSCSI) host, and you want to add an iSCSI port to the host, refer to the **addhostport** and **chhost** commands.

An invocation example

```
rmhostiogrp -iogrp 1:2 host0
```

The resulting output:

```
No feedback
```

rmhostport

Use the **rmhostport** command to delete worldwide port names (WWPNs) or Internet Small Computer System Interface (iSCSI) names from an existing host object.

Syntax

```
►►── rmhostport ── ─┬─ -saswwpn ── wwpn_list ───────┬─┬──────────┬─ ─┬─ host_name ─┬──────────────►◄
                    ├─ -fcwwpn ── wwpn_list ────────┤ └─ -force ─┘   └─ host_id ───┘
                    └─-iscsiname──iscsi_name_list──┘
```

Parameters

-saswwpn *wwpn_list*
(Required if you do not use **iscsiname** or **fcwwpn**) Specifies the colon-separated list of Serial Attached SCSI (SAS) WWPNs with a 16-character hexadecimal string.

-fcwwpn *wwpn_list*
(Required if you do not use **iscsiname** or **saswwpn**) Specifies the colon-separated list of Fibre Channel (FC) WWPNs with a 16-character hexadecimal string.

-iscsiname *iscsi_name_list*
(Required if you do not use **fcwwpn** or **saswwpn**) Specifies the comma-separated list of iSCSI names to delete from the host. At least one WWPN or iSCSI name must be specified. You cannot use this parameter with the **fcwwpn** or **saswwpn** parameter.

-force
(Optional) Overrides the check that specifies all of the WWPNs or iSCSI names in the list are mapped to the host. Ports not associated with the host are ignored.

Important: Using the force parameter might result in a loss of access. Use it only under the direction of your product support information.

host_name | *host_id*
(Required) Specifies the host name or the host ID.

-force
(Optional) Overrides the check that specifies all of the WWPNs or iSCSI names in the list are mapped to the host. Ports that are not associated with the host are ignored.

Important: Using the force parameter might result in a loss of access. Use it only under the direction of your product support information.

Description

This command deletes the list of host-bus adapter (HBA) WWPNs or iSCSI names from the specified host object. If the WWPN ports are still logged in to the fabric, they become unconfigured and are listed as candidate WWPNs.

Any volumes that are mapped to this host object are automatically unmapped from the ports.

Remember: This command is unsuccessful if:

- Volume protection is enabled (using the **chsystem** command)
- The last host port being deleted is mapped to any volume that has received I/O within the defined volume protection time period

If multiple hosts are mapped to the same active volume, the host port removal is allowed if the host is offline. This allows for the removal of ports from hosts that might be part of the same system.

List the candidate Fibre Channel (FC) or serial-attached SCSI (SAS) ports by issuing the **lsfcportcandidate** or **lssasportcandidate** command. A list of the ports that are available to be added to host objects is displayed. To list the WWPNs that are currently assigned to the host, issue the following:

```
lshost hostobjectname
```

where *hostobjectname* is the name of the host object.

Add the new ports to the existing host object by issuing the following command:

```
addhostport -fcwwpn one or more existing WWPNs
separated by : hostobjectname/ID
```

where one or more existing WWPNs separated by colon (:) and *hostobjectname/id* is the name or ID of the host object.

Remove the old ports from the host object by issuing the following command:

```
rmhostport -fcwwpn one or more existing WWPNs
separated by : hostobjectname/ID
```

where one or more existing WWPNs separated by colon (:) corresponds with those WWPNs that are listed in the previous step. Any mappings that exist between the host object and volumes are automatically applied to the new WWPNs.

An invocation example

```
rmhostport -saswwpn 210100E08B251DD4 host1
```

The resulting output:

```
No feedback
```

An invocation example

```
rmhostport -fcwwpn 210100E08B251EE6 host1
```

The resulting output:

```
No feedback
```

An invocation example

```
rmhostport -iscsiname iqn.localhost.hostid.7f000001 mchost13
```

The resulting output:

```
No feedback
```

Chapter 18. Information commands

Use the the information commands to display specific types of system information.

Information commands return no output but exit successfully when there is no information to display.

Important: IDs are assigned at run-time by the system and might not be the same IDs that are used after configuration restoration. Use object names instead of IDs whenever possible.

ls2145dumps (Deprecated)

The **ls2145dumps** command is deprecated. Use the **lsdumps** command to display a list of files in a particular dumps directory.

lsconfigdumps (Discontinued)

The **lsconfigdumps** command is discontinued. Use the **lsdumps** instead.

lssshkeys (Discontinued)

Attention: The **lssshkeys** command is discontinued. Use the user management commands to configure remote authentication service and manage users and user groups on the cluster.

Chapter 19. Livedump commands

Use the **livedump** commands to manage the node livedumps on your system.

cancellivedump

Use the **cancellivedump** command to cancel a live dump.

Syntax

```
►►── cancellivedump ── ──┬─node_name─┬──────────────────────────────────────────►◄
                         └─node_id───┘
```

Parameters

node_name|*node_id*
(Required) Identifies the node name or ID.

Description

Use this command if you issue a **preplivedump** command, but then decide not to issue a **triggerlivedump** command. This releases the resources you allocated for the livedump. This event is recorded in the node trace (.trc) file. For this command to succeed, the node must be in a livedump prepared state.

An invocation example

```
cancellivedump node1
```

The resulting output:

```
No feedback
```

lslivedump

Use the **lslivedump** command to query the livedump state of a node.

Syntax

```
►►── lslivedump ── ──┬─node_name─┬──────────────────────────────────────────────►◄
                     └─node_id───┘
```

Parameters

node_name|*node_id*
(Required) Identifies the node name or ID.

Description

You can issue this command repeatedly to determine if a live dump is in progress for the node. Table 87 on page 514 provides the possible values that are applicable to the attributes that are displayed as data in the output views.

Table 87. lslivedump outputs

Attribute	Description
inactive	The node has no live dump activity.
prepared	The node is ready to be triggered.
dumping	The node is writing the dump file.

An invocation example

```
lslivedump node1
```

The resulting output:

```
status
prepared
```

preplivedump

Use the **preplivedump** command to reserve the system resources that are required for livedump.

Syntax

```
►►— preplivedump — ┬node_name┬——————————————————————————————————————►◄
                   └node_id——┘
```

Parameters

node_name|*node_id*
: (Required) Identifies the node name or ID.

Description

You can prepare more than one node for livedump at a time by issuing the **preplivedump** command consecutively. However, you can only trigger one livedump at a time, with an automatic lag time of 30 seconds between each trigger event. This helps maintain node stability.

You can issue multiple **preplivedump** commands on the same node; however, only a **preplivedump** command followed by a **triggerlivedump** command results in output.

Because the livedump resource allocation can take time to execute, you can issue this command to prepare the livedump but trigger it at a later time. This command times out after 60 seconds. The preplivedump event is located in the node trace (.trc) file.

An invocation example

```
preplivedump node1
```

The resulting output:

```
No feedback
```

triggerlivedump

Use the **triggerlivedump** command to capture the metadata that you want to dump, and write the dump file to the internal disk on the node.

Syntax

```
►►── triggerlivedump ── ──┬─node_name─┬──────────────────────────────────────────────►◄
                          └─node_id───┘
```

Parameters

node_name|*node_id*
(Required) Identifies the node name or ID.

Description

You can issue this command to trigger a **livedump** command. Only one **triggerlivedump** action can be in progress at one time, with an automatic lag time of 30 seconds between each trigger event. The node must have a live dump state of `prepared` for this command to succeed. Output is recorded in the node trace (.trc) file.

After you issue the **triggerlivedump** command, the command captures data and returns you to the CLI interface so that you can issue more commands. While you issue more commands, the live dump disk file is written to the disk in the background, and the live dump state shows as `dumping`. After the write is complete, the state shows as `inactive`.

An invocation example

```
triggerlivedump node1
```

The resulting output:

```
No feedback
```

Chapter 20. Managed disk commands

Use the managed disk commands to work with managed disk options on a system.

If the system detects an MDisk, it automatically adds it to the list of known MDisks. If you delete the array that corresponds to the MDisk, the system only deletes the MDisk from the list if the MDisk is offline and it has a mode of `unmanaged` (it does not belong to a storage pool).

addmdisk

Use the **addmdisk** command to add one or more managed disks to an existing storage pool.

Syntax

```
►►── addmdisk ── ── -mdisk ──┬─ mdisk_id_list ───┬──┬─────────────────────────────────┬── ──────────────►
                             └─ mdisk_name_list ─┘  └─ -tier ──┬─ tier0_flash ──────┬─┘
                                                               ├─ tier1_flash ──────┤
                                                               ├─ tier_enterprise ──┤
                                                               └─ tier_nearline ────┘

►──┬─ mdisk_group_id ───┬──────────────────────────────────────────────────────────────────────────►◄
   └─ mdisk_group_name ─┘
```

Parameters

-mdisk *mdisk_id_list* | *mdisk_name_list*
(Required) Specifies one or more managed disk IDs or names to add to the storage pool.

-tier *tier0_flash* | *tier1_flash* | *tier_enterprise* | *tier_nearline*
(Optional) Specifies the tier of the MDisk or MDisks being added. Unless otherwise specified, the current tier value associated with the MDisk is retained. The values are:

tier0_flash
Specifies a `tier0_flash` hard disk drive or an external MDisk for the newly discovered or external volume.

tier1_flash
Specifies an `tier1_flash` (or flash drive) hard disk drive or an external MDisk for the newly discovered or external volume.

tier_enterprise
Specifies a `tier_enterprise` hard disk drive or an external MDisk for the newly discovered or external volume.

tier_nearline
Specifies a `tier_nearline` hard disk drive or an external MDisk for the newly discovered or external volume.

The default value for a newly discovered unmanaged MDisk is *enterprise*. You can change this value by using the **chmdisk** command.

The tier of external managed disks is not detected automatically and is set to `enterprise`. If the external managed disk is made up of flash drives or nearline Serial Attached SCSI (SAS) drives and you want to use Easy Tier, you must either specify the tier when adding the managed disk to the storage pool or use the **chmdisk** command.

mdisk_group_id | *mdisk_group_name*

(Required) Specifies the ID or name of the storage pool to add the disks to. When an MDisk is added, the warning threshold for the storage pool is automatically scaled.

Description

This command adds the managed disks that you specify to the storage pool.

If there are no MDisks in the storage pool, the site of the MDisk being added must be well-defined. If there are MDisks in the storage pool, the site information for an MDisk being added to a storage pool with HyperSwap or stretched topology system must match the topology of other MDisks in the storage pool.

Remember: This command cannot be used for child pools.

The disks can be specified in terms of the managed disk ID or the managed disk name. The managed disks must be in unmanaged mode.

Disks that already belong to a storage pool cannot be added to another storage pool until they have been deleted from their current storage pool. You can delete a managed disk from a storage pool under the following circumstances:

- If the managed disk does not contain any extents in use by a volume
- If you can first migrate the extents in use onto other free extents within the storage pool.

Remember: Do not include an Mdisk in an storage pool if it can only be used in image mode.

If the system has I/O groups that are not capable of encryption, you cannot add the MDisk if the MDisk group has an encryption key and the MDisk is not self-encrypting.

An invocation example

```
addmdisk -mdisk mdisk13:mdisk14 -tier tier_nearline Group0
```

The resulting output:

```
No feedback
```

applymdisksoftware (Discontinued)

Attention: The **applymdisksoftware** command has been discontinued. Use the **applydrivesoftware** command to update drives.

chmdisk

Use the **chmdisk** command to modify the name or IBM Easy Tier settings for a managed disk (MDisk).

Syntax

```
►►— chmdisk — ┬─────────────────────────┬ ┬──────────────────────────────────┬──────────────────►
              └─ -name — new_name_arg ─┘ └─ -tier ──┬─ tier0_flash ──────┬┘
                                                     ├─ tier1_flash ──────┤
                                                     ├─ tier_enterprise ─┤
                                                     └─ tier_nearline ───┘
```

```
►─┬───────────────────────────────────┬─┬─────────────────────────┬─┬─ mdisk_id ───┬──────────────►◄
  └─ -easytierload ─┬─default───┬─────┘ └─ -encrypt ──┬─ yes ─┬─┘ └─ mdisk_name ─┘
                    ├─low───────┤                    └─ no ──┘
                    ├─medium────┤
                    ├─high──────┤
                    └─very_high─┘
```

Parameters

-name *new_name_arg*
(Optional) Specifies the new name to be applied to the managed disk.

-tier *tier0_flash* | *tier1_flash* | *tier_enterprise* | *tier_nearline*
(Optional) Specifies the new tier of the MDisk. The values are:

tier0_flash
Specifies a tier0_flash hard disk drive or an external MDisk for the newly discovered or external volume.

tier1_flash
Specifies an tier1_flash (or flash drive) hard disk drive or an external MDisk for the newly discovered or external volume.

tier_enterprise
Specifies a tier_enterprise hard disk drive or an external MDisk for the newly discovered or external volume.

tier_nearline
Specifies a tier_nearline hard disk drive or an external MDisk for the newly discovered or external volume.

-easytierload *default* | *low* | *medium* | *high* | *very_high*
(Optional) Specifies the Easy Tier load (amount) to place on a non-array MDisk within its tier.

If Easy Tier is either overusing or under-utilizing a particular MDisk, modify the easy_tier_load value to change the load size.

Note: Specifying *default* returns the performance capability to the value used by the system. Specify *very_high* only if the MDisk tier is ssd.

-encrypt *yes* | *no*
(Optional) Specifies whether the MDisk is encrypted by using its own encryption resources. The values are yes or no.

Important: If you use SAN Volume Controller in front of an encrypted Storwize V7000 system, you must upgrade Storwize V7000 before you apply encryption to your Storwize V7000 system.
If you apply encryption to your system, you must identify the encrypted MDisks before you apply the encryption. If you specify chmdisk -encrypt, the setting is permanent in SAN Volume Controller no matter what Storwize V7000 says.

mdisk_id | *mdisk_name*
(Required) Specifies the ID or name of the managed disk to modify.

Description

This command modifies the attributes of a managed disk.

Do not use the **-encrypt** parameter if one of the MDisk group's has an encryption key, parent pool, and child pools. Use **chmdisk** for existing self-encrypting MDisks before you start any migration. If an MDisk is self-encrypting, the encrypted property defaults to what is reported.

If you are upgrading your clustered system (system) and the back end of the system uses encrypted storage, you must indicate which MDisks are self-encrypting before you add MDisks to a storage pool. If those MDisks are part of a storage pool, the system assumes that the back-end is not self-encrypting (even if it might be).

If you create encrypted storage pools, the system encrypts locally before it sends data to the back-end. So, the back-end of the system could encrypt again and cannot compress data because the data is random and not compressible.

Note: You must upgrade the system first.
To use encryption on the system that already has encryption that is enabled on the back-end, upgrade the back-end of the system before you enable encryption on the system.

An invocation example

```
chmdisk -tier tier0_flash mdisk13
```

The resulting output:

```
No feedback
```

An invocation example

```
chmdisk -tier tier_nearline mdisk0
```

The resulting output:

```
MDisk Group, id [13], successfully created
```

An invocation example

```
chmdisk -easytierload high mdisk0
```

The resulting output:

```
MDisk Group, id [13], successfully created
```

An invocation example

```
chmdisk -name my_first_mdisk -encrypt yes 0
```

The resulting output:

```
MDisk Group, id [0], successfully changed
```

detectmdisk

Use the **detectmdisk** command to manually rescan the Internet Small Computer Systems Interface (iSCSI) or Fibre Channel (FC) network for any new managed disks (MDisks) that might have been added, and to rebalance MDisk access across all available controller device ports.

Syntax

```
►►— detectmdisk — -scope — scope_id ————————————————————————————————►◄
```

Parameters

-scope *scope_id*
(Optional) Specifies the domain index. The value must be a number from 0 to 6 For example, the value 0 indicates FC and 6 indicates iSCSI.

Description

This command causes the clusterd system (system) to rescan the FC or iSCSI network. The rescan discovers any new MDisks that have been added to the system and rebalances MDisk access across the available controller device ports. This command also detects any loss of controller port availability, and updates the SAN Volume Controller configuration to reflect any changes.

Note: Although it might appear that the **detectmdisk** command has completed, some extra time might be required for it to run. The **detectmdisk** is asynchronous and returns a prompt while the command continues to run in the background. You can use the **lsdiscoverystatus** command to list the discovery status.

In general, the system automatically detects disks when they appear on the network. However, some FC controllers do not send the required SCSI primitives that are necessary to automatically discover the new disks.

If you have attached new storage and the system has not detected it, you might need to run this command before the system detects the new disks.

When back-end controllers are added to the FC SAN and are included in the same switch zone as a system, the system automatically discovers the back-end controller and determines what storage is presented to it. The SCSI LUs that are presented by the back-end controller are displayed as unmanaged MDisks. However, if the configuration of the back-end controller is modified after this has occurred, the system might be unaware of these configuration changes. Run this command to rescan the FC or iSCSI network and update the list of unmanaged MDisks.

Note: The automatic discovery that is performed by the system does not write to an unmanaged MDisk. Only when you add an MDisk to a storage pool, or use an MDisk to create an image mode volume, is the storage actually used.

To identify the available MDisks, issue the **detectmdisk** command to scan the FC or iSCSI network for any MDisks. When the detection is complete, issue the **lsmdiskcandidate** command to show the unmanaged MDisks; these MDisks have not been assigned to a storage pool. Alternatively, you can issue the **lsmdisk** command to view all of the MDisks.

If disk controller ports have been removed as part of a reconfiguration, the SAN Volume Controller detects this change and reports the following error because it cannot distinguish an intentional reconfiguration from a port failure:

```
1630 Number of device logins reduced
```

If the error persists and redundancy has been compromised, the following more serious error is reported:

```
1627 Insufficient redundancy in disk controller connectivity
```

You must issue the **detectmdisk** command to force the SAN Volume Controller to update its configuration and accept the changes to the controller ports.

Note: Only issue the **detectmdisk** command when all of the disk controller ports are working and correctly configured in the controller and the SAN zoning. Failure to do this could result in errors not being reported.

An invocation example

```
detectmdisk
```

The resulting output:

```
No feedback
```

An invocation example

```
detectmdisk -scope 1
```

The resulting output:

```
No feedback
```

dumpallmdiskbadblocks

Use the **dumpallmdiskbadblocks** command to dump bad block counts to a dump file used by the fix procedures and the satask **snap** command.

Syntax

```
►►── dumpallmdiskbadblocks ── ──────────────────────────────────────────────►◄
```

Parameters

None

Description

Use the **dumpallmdiskbadblocks** command to dump bad block counts to a readable ASCII dump file for use by fix procedures and the satask **snap** command. The output contains bad blocks for which an error log has been raised.

Use **lsdumps -prefix** `/dumps/mdisk` to list the output files. Use **cleardumps -prefix** `/dumps/mdisk` to clear the output files.

The maximum number of dump files is 20.

An invocation example

```
dumpallmdiskbadblocks
```

The resulting output if MDisk 2 and MDisk 5 have bad blocks:

```
Cluster name: my_cluster
Timestamp of dump: Fri Oct 31 11:27:33 2009 UTC

Mdisk id: 2
Mdisk name: mdisk2
Number of bad blocks: 4

Mdisk id: 5
Mdisk name: mdisk 5
Number of bad blocks: 1

Total mdisks with bad blocks: 2
Total number of bad blocks: 5
```

The resulting output if the MDisks have no bad blocks

```
Cluster name: my_cluster
Timestamp of dump: Fri Oct 31 11:27:33 2009 UTC

Total mdisks with bad blocks: 0
Total number of bad blocks: 0
```

dumpmdiskbadblocks

Use the **dumpmdiskbadblocks** command to write the bad block counts and locations that are on a specified MDisk to a dump file for use by fix procedures.

Syntax

```
►►── dumpmdiskbadblocks ── ─┬─ object_id ───┬──────────────────────────────────────────────►◄
                            └─ object_name ─┘
```

Parameters

object_id | *object_name*
(Required) Specifies the MDisk for which you need to dump the bad block record table.

Description

Use the **dumpmdiskbadblocks** command to write the bad block counts and locations that are on a specified MDisk to a readable ASCII dump file for use by fix procedures. The output consists of bad blocks for which an event log has been raised.

Use **lsdumps -prefix** /dumps/mdisk to list the output files. Use **cleardumps -prefix** /dumps/mdisk to clear the output files.

The reported event log sequence numbers correspond to the first event seen in the bad block record, which is a 512-block region.

- If there are multiple event logs in the same region, the earliest event sequence is used.
- If there are event logs of different types in the same region, event sequence numbers for bad blocks caused by medium errors on RAID member drives take precedence.
- If a range of bad blocks runs across record boundaries, the sequence number corresponding to the last record is used.

The maximum number of dump files is 20.

An invocation example

```
dumpmdiskbadblocks 3
```

The resulting output if the MDisk has bad blocks:

```
Cluster name: my_cluster
Timestamp of dump: Fri Oct 31 11:27:33 2017 UTC

Mdisk id: 3
Mdisk name: mdisk3
Number of bad blocks: 6

Start LBA: 0x1234123412341234
Length: 2
Event log sequence number: 1

Start LBA: 0x5678568102341234
Length: 4
Event log sequence number: 2
```

The resulting output if the MDisk has no bad blocks:

```
Cluster name: my_cluster
Timestamp of dump: Fri Oct 31 11:27:33 2017 UTC

Mdisk id: 3
Mdisk name: mdisk3
Number of bad blocks: 0
```

includemdisk

Use the **includemdisk** command to include a disk that has been excluded by the clustered system (system).

Syntax

```
►►─ includemdisk ─ ─┬─ mdisk_id ───┬──────────────────────────────────────────────────►◄
                    └─ mdisk_name ─┘
```

Parameters

mdisk_id | *mdisk_name*
(Required) Specifies the ID or name of the managed disk to add back into the system.

Description

The specified managed disk is included in the system.

You might exclude a disk from the system because of multiple I/O failures. These failures might be caused by noisy (or unstable) links. Once a fabric-related problem has been fixed, the excluded disk can be added back into the system.

Running this command against an MDisk might change its state, whether the state is reported as excluded.

Note: If an MDisk is in the excluded state, is offline, and does not belong to an storage pool, issuing an include command for this MDisk results in the MDisk record being deleted from the system.

An invocation example

```
includemdisk mdisk5
```

The resulting output:

```
No feedback
```

lsmdisk

Use the **lsmdisk** command to display a concise list or a detailed view of managed disks (MDisks) visible to the clustered system (system). It can also list detailed information about a single MDisk.

Syntax

```
►►─ lsmdisk ─ ─┬───────────────────────────────────────────────────────┬─ ──────────────►
               └─ -filtervalue ─ attribute=value ─ ─┬────────────────┬─┘
                                                    └─ -unit ─┬─ b ─┬─┘
                                                              ├─ kb ─┤
                                                              ├─ mb ─┤
                                                              ├─ gb ─┤
                                                              ├─ tb ─┤
                                                              └─ pb ─┘
```

```
►─┬─────────────────┬──┬──────────┬──┬──────────┬──┬───────────────────────┬──────────────────►
  └─ -filtervalue? ─┘  └─ -nohdr ─┘  └─ -bytes ─┘  └─ -delim ── delimiter ─┘

►─┬─────────────────┬──────────────────────────────────────────────────────────────────────────►◄
  ├─ object_id ─────┤
  └─ object_name ───┘
```

Parameters

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filter attributes that match the specified values; see **-filtervalue?** for the supported attributes. Only objects with a value that matches the filter attribute value are returned. If **capacity** is specified, the units must also be included. Use the **unit** parameter to interpret the value for size or capacity.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards with the system CLI:

- The wildcard character is an asterisk (*).
- The command can contain a maximum of one wildcard, which must be the first or last character in the string.
- When you use a wildcard character, you must enclose the filter entry within double quotation marks (""), as follows:

```
lsmdisk -filtervalue  "name=md*"
```

-unit b | kb | mb | gb | tb | pb
(Optional) Specifies the data units for the **-filtervalue** parameter.

Note: **-unit** must be used with **-filtervalue**.

-filtervalue?
(Optional) Includes all of the valid filter attributes in the report. The following filter attributes are valid for the **lsmdisk** command:

- block_size
- capacity
- controller_id
- controller_name
- ctrl_LUN_#
- easy_tier_load
- id
- max_path_count
- mode
- mdisk_grp_id
- mdisk_grp_name
- name
- path_count
- quorum_index
- site_id
- site_name
- status
- tier
- UID

Any parameters that are specified with the **-filtervalue?** parameter are ignored.

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-bytes
(Optional) Specifies that you want the report to display all capacities as bytes. Capacity values that are displayed in units other than bytes might be rounded. When you filter on capacity, use a unit of bytes, **-unit b**, for exact filtering.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space that is separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim **:** on the command line, the colon character (**:**) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

object_id **|** *object_name*
(Optional) Specifies the name or ID of an object. When you use this parameter, the detailed view of the specific object is returned and any value that is specified by the **-filtervalue** parameter is ignored. If you do not specify the *object_id* | *object_name* parameter, the concise view displays all objects that match the filtering requirements that are specified by the **-filtervalue** parameter.

Description

This command returns a concise list or a detailed view of MDisks visible to the system. Table 88 provides the potential output for MDisks.

Note: Some of the attributes may not be applicable to your system.

Table 88. MDisk output

Attribute	Values
status	- online - offline - excluded - degraded_paths - degraded_ports - degraded (This value applies only to internal MDisks.)
mode	unmanaged, managed, image, array
quorum_index	0, 1, 2, or blank if the MDisk is not being used as a quorum disk.
block_size	512, 524 bytes in each block of storage
ctrl_type	4, 6, where 6 is a flash drive that is attached inside a node and 4 is any other device.

Table 88. MDisk output (continued)

Attribute	Values
tier	The tier this MDisk is assigned to by auto-detection (for internal MDisks) or by the user: • `tier0_flash` • `tier1_flash` • `tier_enterprise` • `tier_nearline` **Note:** You can change this value by using the **`chmdisk`** command.
easy_tier_load	This value controls Easy Tier settings, and is either blank (for arrays) or one of the following values (for MDisks): • `low` • `medium` • `high` • `very_high`
raid_status	**offline** Array is offline on all nodes. **degraded** Array has deconfigured or offline members; the array is not fully redundant. **syncing** Array members are all online. The array is syncing parity or mirrors to achieve redundancy. **initting** Array members are all online. The array is initializing; the array is fully redundant. **online** Array members are all online, and the array is fully redundant.
raid_level	The RAID level of the array (`RAID0`, `RAID1`, `RAID5`, `RAID6`, `RAID10`).
redundancy	The number of how many member disks that fail before the array fails.
strip_size	The strip size of the array (in KB).
spare_goal	The number of spares that the array members must be protected by.
spare_protection_min	The minimum number of spares that an array member is protected by.
balanced	Describes if the array is balanced to its spare goals: **exact** All populated members have exact capability match, exact location match. **yes** All populated members have at least exact capability match, exact chain, or different enclosure or slot. **no** Anything else that is not included for `yes` or `exact`.
site_id	Indicates the site value for the MDisk. This numeric value is `1`, `2`, `3`, or blank.
site_name	Indicates the site name for the MDisk. This is an alphanumeric value or is blank.
fabric_type	Indicates the type of MDisk. The values are: • `fc` indicates that it is an MDisk from an FC controller. • `sas_direct` indicates that it is an MDisk from an SAS direct-attached controller. • `iscsi` indicates that it is an iSCSI controller.

Table 88. MDisk output (continued)

Attribute	Values
encrypt	Indicates whether the data stored on the MDisk group is encrypted or not encrypted. The values are: • yes indicates that the pool has an encryption key. • no indicates that the pool does not have an encryption key but it contains an MDisk or MDisks that are encrypted. • Blank if the pool doesn't have an encryption key and the pool has no MDisks.
distributed	Indicates whether the array is distributed. The values are yes or no.
drive_class_id	Indicates the drive class that makes up this array. If **-allowsuperior** was used during array creation, the lowest used drive class ID is displayed. This value is blank for traditional arrays.
drive_count	Indicates the total width of the array, including rebuild areas. The value is a number in the range 4 - 128. The minimum value for RAID-6 and RAID-10 arrays is 6.
stripe_width	Indicates the width of a single unit of redundancy within a distributed set of drives. The values are: • Any number in the range 3 - 16 for RAID-5 arrays. • Any number in the range 4 - 16 for RAID-6 arrays. • An even number in the range 2 - 16 for RAID-10 arrays.
rebuild_areas_total	Indicates the total number of rebuild areas set at array creation time. These rebuild areas provide performance but no capacity. The value is a number in the range 1 - 4 for distributed array RAID-5 and RAID-6, and the value is a number in the range 2 - 4 for distributed array RAID-10 (the value is blank for traditional arrays).
rebuild_areas_available	Indicates the number of remaining build areas within the set of arrays. The value is a number in the range 1 - 4 for distributed array RAID-5 and RAID-6, and the value is a number in the range 2 - 4 for distributed array RAID-10 (the value is blank for traditional arrays).
rebuild_areas_goal	Indicates the rebuild areas threshold (minimum limit) at which point the array logs an error. The value is a number in the range 1 - 4 for distributed array RAID-5 and RAID-6, and a number in the range 2 - 4 for distributed array RAID-10 (the value is blank for traditional arrays).
dedupe	Indicates that dedupe is enabled. If dedupe is enabled, duplicate copies of repeating data are compressed or removed.
ctrl_WWNN	Indicates the control worldwide node name (WWNN).
preferred_WWPN	Indicates the preferred worldwide port name (WWPN).
active_WWPN	Indicates the active WWPN.
preferred_iscsi_port_id	Indicates the preferred I/O port identifier, which has the same value as the preferred_WWPN value in the Fibre Channel (FC) domain. The Internet Small Computer System Interface (iSCSI) port ID value is displayed, but the value is blank for non-iSCSI domains. This value must be a numeric value that can range in the range 0 - 1023.
active_iscsi_port_id	Indicates the active I/O port identifier, which has the same value as the active_WWPN value in the FC domain. The Internet Small Computer System Interface (iSCSI) port ID value is displayed, but the value is blank for non-iSCSI domains. This value must be a numeric value that can range in the range 0 - 1023.
over_provisioned	Indicates whether the MDisk is thin-provisioned. The value is no if the MDisk is marked as fully allocated. resource-provisioned, the information cannot be verified at the backend. The value is yes or no.

Table 88. MDisk output (continued)

Attribute	Values
supports_unmap	Indicates whether the mdisk is provided by a controller that indicates that it supports unmapping. The value is `no` if the MDisk indicates that it does not support unmapping or this information cannot be verified at the backend. The value is `yes` or `no`.
provisioning_group_id	Indicates the allocated identifier for the provisioning group affiliated with the MDisk. The identifier lists the MDisks that are contained in the same provisioning group. The value must be an integer (number).
physical_capacity	Indicates the total physical storage capacity of the provisioning group that contains this MDisk. If this MDisk is not over-provisioned the logical capacity is displayed here. The value must be a number (indicated in units) that is rounded to two decimal places.
physical_free_capacity	Indicates the amount of formatted available physical space in the provisioning group that contains this MDisk. If this MDisk is not over-provisioned the remaining logical capacity is displayed instead. The value must be a number (indicated in units) that is rounded to two decimal places.

Note: The automatic discovery that is performed by the system does not write anything to an unmanaged MDisk. It is only when you add an MDisk to a storage pool, or use an MDisk to create an image mode volume, that the system uses the storage.

To see which MDisks are available, issue the **`detectmdisk`** command to manually rescan the Fibre Channel or iSCSI network for any new MDisks. Issue the **`lsmdiskcandidate`** command to show the unmanaged MDisks. These MDisks are not assigned to a storage pool.

Notes:

1. A system connection from a node or node canister port to a storage controller port for a single MDisk is a path. The Mdisk *path_count* value is the number of paths currently being used to submit input/output (I/O) to this MDisk.
2. The MDisk *max_path_count* value is the highest value *path_count* reaches since the MDisk was last fully online.
3. The *preferred_WWPN* is one of the World Wide Port Names (WWPNs) the storage controller specifies as a preferred WWPN. If the controller has nothing that is specified, this is a blank field.
4. The *active_WWPN* indicates the WWPN of the storage controller port currently being used for I/O.
   a. If no storage controller ports are available for I/O, this is a blank field.
   b. If multiple controller ports are actively being used for I/O, this field's value is `many`.

The following define the status fields:

online The MDisk is online and available.

degraded
(Internal MDisks only) The array has members that are `degraded`, or the `raid_status` is `degraded`.

degraded_ports
There are one or more MDisk port errors.

degraded_paths
One or more paths to the MDisk are lost; the MDisk is not online to every node in the system.

offline
All paths to the MDisk are lost.

excluded

The MDisk is excluded from use by the system; the MDisk port error count exceeded the threshold.

A concise invocation example

```
lsmdisk -delim :
```

The concise resulting output:

```
id:name:status:mode:mdisk_grp_id:mdisk_grp_name:capacity:ctrl_LUN_#:controller_name:UID:tier:encrypt:site_id:site_name:dist
0:mdisk0:online:managed:2:Storwize:200.0GB:0000000000000000:controller0:6005076d0281003d2000000000000043e0000000000000000000
6:mdisk6:online:managed:1:A9000:192.5GB:0000000000000002:controller2:6001738cfc900cef000000000001348e0000000000000000000000
```

A detailed invocation example

```
lsmdisk mdisk1
```

The detailed resulting output:

```
id:1
name:mdisk1
status:online
mode:array
mdisk_grp_id:0
mdisk_grp_name:mdgp0
capacity:136.0GB
quorum_index:
block_size:512
controller_name:controller1
ctrl_type:4
ctrl_WWNN:200400A0B80F0702
controller_id:1
path_count:2
max_path_count:2
ctrl_LUN_#:0000000000000002
UID:600a0b80000f07020000005c45ff8a7c00000000000000000000000000000000
preferred_WWPN:200400A0B80F0703
active_WWPN:200400A0B80F0703
fast_write_state:empty
raid_status:
raid_level:
redundancy:
strip_size:
spare_goal:
spare_protection_min:
balanced:
tier:tier0_flash
slow_write_priority:latency
fabric_type:fc
site_id:2
site_name:2
easy_tier_load:low
encryt:no
distributed:no
 drive_class_id
 drive_count:8
 stripe_width:4
 total_rebuild_areas
 available_rebuild_areas
 rebuild_areas_goal
 preferred_iscsi_port_id
 active_iscsi_port_id
dedupe:no

flashsystem no
over_provisioned:no
```

```
supports_unmap no
provisioning_group_id
physical_capacity
physical_free_capacity
```

lsmdiskdumps (Deprecated)

The **lsmdiskdumps** command is deprecated. Use the **lsdumps** command to display a list of files in a particular dumps directory.

lsmdisklba

Use the **lsmdisklba** command to list the MDisk and logical block address (LBA) for the specified volume LBA.

Syntax

```
►►— lsmdisklba — — -vdisklba — vdisklba — ┬───────────────┬─ ┬──────────────────────┬─ ───────►
                                          └ -copy— id ─┘   └ -delim — delimiter ─┘

►┬──────────┬─ — -vdisk ─┬ vdisk_id ───┬──────────────────────────────────────────────────►◄
 └ - nohdr─┘             └ vdisk_name ─┘
```

Parameters

-vdisklba *vdisklba*
(Required) Specifies the 64–bit hexadecimal logical block address (LBA) on the volume. The LBA must be specified in hex, with a 0x prefix.

-copy *id*
(Optional) Specifies the volume copy ID to list the MDisk and LBA for. If this parameter is not specified, the command lists MDisks and LBAs for all volume copies.

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space that is separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim : on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

vdisk_id | *vdisk_name*
(Required) Specifies the volume name or ID.

Description

The **lsmdisklba** command returns the logical block address (LBA) of the MDisk that is associated with the volume LBA. For mirrored volume, the command lists the MDisk LBA for both the primary and the copy.

If applicable, the command also lists the range of LBAs on both the volume and MDisk that are mapped in the same extent, or for thin-provisioned disks, in the same grain. If a thin-provisioned volume is offline and the specified LBA is not allocated, the command displays the volume LBA range only.

The mdisk_lba field provides the corresponding LBA on the real capacity for the input LBA. For compressed volume copies it is empty, and the system displays only the range of physical LBAs where the compressed input LBA is located.

Table 89 summarizes the data that can be returned with this command.

Table 89. lsmdisklba command output

			Mirrored volume with one normal copy and one offline thin-provisioned copy	
Field	**Fully allocated, single copy volume**	**LBA not allocated on thin-provisioned volume**	**Normal copy**	**Thin-provisioned copy**
copy_id	yes	yes	yes	yes
mdisk_id	yes	no	yes	no
mdisk_name	yes	no	yes	no
type	allocated	unallocated	allocated	offline
mdisk_lba	yes	no	yes	no
mdisk_start	yes	no	yes	no
mdisk_end	yes	no	yes	no
vdisk_start	yes	yes	yes	yes
vdisk_end	yes	yes	yes	yes

An invocation example

```
lsmdisklba -vdisk 0 -vdisklba 0x123
```

The resulting output:

```
copy_id mdisk_id mdisk_name type      mdisk_lba          mdisk_start        mdisk_end          vdisk_start vdisk_end
0       1        mdisk1     allocated 0x0000000000100123 0x0000000000100000 0x00000000001FFFFF 0x00000000  0x000FFFFF
```

lsmdiskcandidate

Use the **lsmdiskcandidate** command to list all unmanaged MDisks by MDisk ID.

Syntax

```
►►── lsmdiskcandidate ── ─┬──────────┬─ ─┬────────────────────┬──────────────────►◄
                          └─ -nohdr ─┘   └─ -delim ── delimiter ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each

column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (**:**) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

Description

This command displays a list of MDisks that are unmanaged. Only the MDisk IDs are displayed.

When back-end controllers are added to the Fibre Channel SAN and are included in the same switch zone as a cluster, the cluster automatically detects the back-end controller to determine which storage is presented to the node. The SCSI logical units that are presented by the back-end controller are displayed as unmanaged MDisks. However, if the configuration of the back-end controller is modified after this occurs, the cluster might be unaware of these configuration changes. You can then request that the cluster rescan the Fibre Channel SAN to update the list of unmanaged MDisks.

Note: The automatic detection that is performed by the cluster does not write anything to an unmanaged MDisk. It is only when you instruct the cluster to add an MDisk to a storage pool or use a MDisk to create an image mode volume that the storage is used.

Check to see which MDisks are available by issuing the **detectmdisk** command to manually scan the Fibre Channel network for any MDisks. Issue the **lsmdiskcandidate** command to show the unmanaged MDisks. These MDisks are not assigned to a storage pool. Alternatively, you can issue the **lsmdisk** command to view all of the MDisks.

An invocation example

```
lsmdiskcandidate
```

The resulting output:

```
id
5
6
7
8
9
10
11
12
13
14
```

lsmdiskextent

Use the **lsmdiskextent** command to display the extent allocation between managed disks and volumes. The output lists a volume ID, volume copy ID, and the number of extents.

Syntax

```
►►── lsmdiskextent ── ──────────── ────────────────────── ──┬─ mdisk_name ─┬──────────────►◄
                       └─ -nohdr ─┘  └─ -delim ── delimiter ─┘   └─ mdisk_id ───┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If no data exists to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

mdisk_name | *mdisk_id*
(Required) Specifies the specific object ID or name of the specified type.

Description

The command displays a list, in which each entry contains a volume ID, volume copy ID, and the number of extents. These volume copies are using extents on the specified MDisk. The number of extents that are being used on each MDisk is also shown.

Note: You cannot specify this command for MDisks that are in data reduction pools. It means that for:

- Thin-provisioned or compressed volumes, the number extents that are shown is not accurate.
- Fully allocated volumes, the number of extents that are shown is accurate.

A thin-provisioned or compressed volume in a data reduction pool cannot display how many extents are on an MDisk that is in a data reduction pool.

Every volume copy is constructed from one or more MDisks. At times, you might have to determine the relationship between the two objects.

To determine the relationship between volume copies and MDisks, issue the following command for each volume copy:

```
lsvdiskmember vdisk_name | vdisk_id
```

where *vdisk_name* | *vdisk_id* is the name or ID of the volume copy. It displays a list of IDs that correspond to the MDisks that make up the volume copy.

To determine the relationship between volume copies and MDisks and the number of extents that are provided by each MDisk, you must use the command-line interface. For each volume copy, issue the following command:

```
lsvdiskextent vdisk_name | vdisk_id
```

where *vdisk_name* | *vdisk_id* is the name or ID of the volume copy. It displays a table of MDisk IDs and the corresponding number of extents that each MDisk is providing as storage for the specified volume copy.

To determine the relationship between MDisks and volume copies, issue the following command for each MDisk:

```
lsmdiskmember mdisk_name | mdisk_id
```

where *mdisk_name* | *mdisk_id* is the name or ID of the MDisk. It displays a list of IDs that correspond to the volume copies that are using this MDisk.

To determine the relationship between MDisks and volume copies and the number of extents that are used by each volume copy, you must use the command-line interface. For each MDisk, issue the following command:

```
lsmdiskextent mdisk_name | mdisk_id
```

where *mdisk_name* | *mdisk_id* is the name or ID of the MDisk. This command displays a table of volume copy IDs and the corresponding number of extents that are being used by each volume copy. In the output, *number_of_extents* displays either a number (for fully allocated volumes in data reduction pools or volumes in regular pools) or a 1 (for thin-provisioned/compressed volumes in data reduction pools).

An invocation example

```
lsmdiskextent -delim : mdisk0
```

The resulting output:

```
id:number_of_extents:copy_id
1:1:1
```

lsmdiskmember

Use the **lsmdiskmember** command to display a list of volumes that use extents on the specified MDisk. That is, the volumes use extents on the managed disk that are specified by the MDisk ID.

Syntax

```
►►── lsmdiskmember ── ┬──────────┬─ ┬─────────────────────┬─ ┬─ mdisk_id ───┬──────────────►◄
                      └─ -nohdr ─┘  └─ -delim ─ delimiter ─┘  └─ mdisk_name ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If no data exists to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

mdisk_id | *mdisk_name*
(Required) Specifies the ID or name of the MDisk for which you want a list of volumes that use extents of that MDisk.

Description

This command displays a list of volumes that use extents on the managed disk that are specified by the ID. The list displays members of the respective object and is independent of the state of the individual members. That is, if they are in offline state, they are still displayed.

- A volume in a data reduction storage pool cannot display how many members are on an MDisk that is in a data reduction pool.
- If the MDisk specified is in a data reduction pool, output includes all thin-provisioned and compressed volumes in the pool.
- Fully allocated VDisks in data reduction pools are displayed correctly.

Every volume is constructed from one or more MDisks. To determine the relationship between volume copies and MDisks, issue the following command:

```
lsvdiskmember vdisk_id | vdisk_name
```

where *vdisk_id* | *vdisk_name* is the name or ID of the volume copy. This action displays a list of IDs that correspond to the MDisks that make up the volume copy.

To determine the relationship between volume copies and MDisks and the number of extents that are provided by each MDisk, you must use the command-line interface. For each volume copy, issue the following command:

```
lsvdiskextent vdisk_id | vdisk_name
```

where *vdisk_id* | *vdisk_name* is the name or ID of the volume copy. This command displays a table of MDisk IDs and the corresponding number of extents that each MDisk provides as storage for the volume copy.

To determine the relationship between MDisks and volume copies, issue the following command:

```
lsmdiskmember mdisk_id | mdisk_name
```

where *mdisk_id* | *mdisk_name* is the name or ID of the MDisk. This command displays a list of IDs that correspond to the volume copies that are using this MDisk.

To determine the relationship between MDisks and volume copies and the number of extents that are used by each volume copy, you must use the command-line interface. For each MDisk *mdisk_id* | *mdisk_name*, issue the following command:

```
lsmdiskextent mdisk_id | mdisk_name
```

where *mdisk_id* | *mdisk_name* is the name or ID of the MDisk. This command displays a table of volume copy IDs and the corresponding number of extents that are being used by each volume copy.

An invocation example

```
lsmdiskmember -delim : 1
```

The resulting output:

```
id:copy_id
0:0
1:0
2:0
```

```
3:0
4:0
5:0
6:0
```

setquorum (Deprecated)

The **setquorum** command is deprecated. Use the **chquorum** command to change the quorum association.

triggermdiskdump (Discontinued)

Attention: The **triggermdiskdump** command is discontinued. Use the **triggerdrivedump** command to collect support data from a disk drive.

Chapter 21. Copy Service commands

Use the Copy Service commands to work with the Metro Mirror, Global Mirror, and HyperSwap services that are provided by the system.

chpartnership

Use the **chpartnership** command to modify the bandwidth of the partnership between the local clustered system (system) and the remote system that is specified in the command. This affects the bandwidth that is available for background copy in a system partnership by either Metro Mirror or Global Mirror operations. Additionally, use this command to disable and re-enable the partnership, which allows the local system to be disconnected and then reconnected to the remote system.

Syntax

```
►►— chpartnership —┬─ -start ─┬──┬─ remote_cluster_id ───┬───────────────────►◄
                   └─ -stop ──┘  └─ remote_cluster_name ─┘


►►— chpartnership —┬──────────────────────┬─ ┬────────────────────────────────┬─ ──────►
                   └─ -type ─┬─ ipv4 ─┬───┘  └─ -clusterip ─┬─ newipv4addr ─┬─┘
                             └─ ipv6 ─┘                     └─ newipv6addr ─┘

►─┬──────────────────────────────────┬─ ┬──────────────────┬─ ──────────────────►
  └─ -chapsecret — newCHAPsecret ────┘  └─ -nochapsecret ──┘

►─┬─────────────────────────────────────┬─ ─ ┬──────────────────────────────────────────────┬─►
  └─ -backgroundcopyrate — percentage ──┘    └─ -linkbandwidthmbits — link_bandwidth_in_mbps ─┘

►─ ┬──────────────────────────────┬─ ┬─ remote_cluster_id ───┬────────────────────►◄
   └─ -compressed ─┬─ yes ─┬──────┘  └─ remote_cluster_name ─┘
                   └─ no ──┘
```

Parameters

-start | -stop
(Optional) Starts or stops a Metro Mirror or Global Mirror partnership. To start or stop a partnership, run the **chpartnership** command from either system.

-type *ipv4* | *ipv6*
(Optional) Specifies the Internet Protocol (IP) address format for the partnership using either of these case-sensitive strings:

- *ipv4* for Internet Protocol Version 4 (IPv4)
- *ipv6* for Internet Protocol Version 6 (IPv6)

This migrates a partnership from *ipv4* to *ipv6* or vice versa.

-clusterip *newipv4addr* | *newipv6addr*
(Optional) Specifies the new partner system IP address, either *ipv4* or *ipv6*. Systems connected over IP links are not displayed by **lspartnershipcandidate** before executing **mkippartnership**. This does not apply to FC-based or FCoE-based connections.

Specify this parameter when creating partnerships with systems connected over native IP links. To change the partner system IP address, first specify **chapartnership -stop** to stop the partnership.

-chapsecret *newCHAPsecret*

(Optional) Specifies the new Challenge-Handshake Authentication Protocol (CHAP) secret of the partner system. The maximum size of the CHAP secret is eighty alphanumeric characters.

-nochapsecret

(Optional) Resets the CHAP secret used to authenticate with the partner system. Specify **chpartnership -stop** to stop the partnership. Reset the CHAP secret of the partner system when authentication of discovery requests is turned off on the partner system (by specifying **chsystem -rcauthmethod**).

-backgroundcopyrate *percentage*

(Optional) Specifies the maximum percentage of aggregate link bandwidth that can be used for background copy operations. This parameter can be specified without stopping the partnership. The percentage is a numeric value from `0` to `100`, and the default value is `50`, which means that a maximum of 50% of the aggregate link bandwidth can be used for background copy operations. This command is mutually-exclusive with all parameters other than **-linkbandwidthmbits**.

Note: If the specified value is non-zero, the combination of both the **-backgroundcopyrate** and the **-linkbandwidthmbits** values must result in a background copy bandwidth of at least 8 megabits per second (Mbps).

-linkbandwidthmbits *link_bandwidth_in_mbps*

(Optional) Specifies the aggregate bandwidth of the RC link between two clustered systems (systems) in megabits per second (Mbps). It is a numeric value from `1` to `100000`, specified in Mbps.

Important: For partnerships over IP links with compression, this parameter specifies the aggregate bandwidth after the compression had been applied to the data. Do not set this parameter higher than the physical link bandwidth multiplied by the (carefully rounded down) compression factor.

This parameter can be specified without stopping the partnership. This command is mutually-exclusive with all parameters other than **-backgroundcopyrate**.

Note: If the specified value is non-zero, the combination of both the **-backgroundcopyrate** and the **-linkbandwidthmbits** values must result in a background copy bandwidth of at least 8 Mbps.

-compressed *yes* | *no*

(Optional) Specifies whether compression is enabled for this partnership. The default value is `no`.

remote_cluster_ID | *remote_cluster_name*

(Required) Specifies the remote system ID or name of a partnership. The specified value must match one of the system IDs or names returned after issuing **lspartnershipcandidate**. The specified value must match one of the system IDs or names listed by **lspartnership**.

Remember: Specifying a remote system ID or name with **chpartnership** does not affect the remote system. To change the system name, specify **chsystem**.

To configure the maximum bandwidth available for Metro Mirror intrasystem relationships, specify:

- A local system ID or name
- The **-linkbandwidthmbps** and **-backgroundcopyrate** parameters

Description

This command modifies the bandwidth of the partnership between the local system and the remote system specified in the command. This affects the bandwidth available for a background copy in Metro Mirror or Global Mirror relationships (from the local to the remote system) . To modify the background copy bandwidth from remote system to local system, issue **chpartnership** a second time for the remote system.

Important: For partnerships over IP links with compression, this parameter specifies the aggregate bandwidth after the compression had been applied to the data. Do not set this parameter higher than the physical link bandwidth multiplied by the (carefully rounded down) compression factor.

Change the CHAP secret or system IP for partnerships created using IP links. Before changing the partner CHAP secret or system IP, stop the partnership.

If a stop partnership is issued, the state is `not_present` briefly before it changes to `fully_configured_stopped`.

Important:

- If you start with a fully configured remote copy partnership, the state (as reported by **`lspartnership`**) is `fully_configured`.
- If a stop partnership is issued, the state is `not_present` (typically for ten seconds or less) before it becomes `fully_configured_stopped`.

After making the necessary changes, start the partnership.

The system partnership must be in either the `partially_configured_stopped` or `fully_configured_stopped` states to be started.

Note: The local and remote systems in an IP partnership must use the same IP address types, IPv4 or IPv6.

An invocation example

```
chpartnership -stop cluster1
```

The resulting output:

```
No feedback
```

An invocation example to change the allocated background copy rate

```
chpartnership -backgroundcopyrate 20 remote-system-2
```

The resulting output:

```
No feedback
```

An invocation example to change the link bandwidth

```
chpartnership -linkbandwidthmbits 1024 remote-system-2
```

The resulting output:

```
No feedback
```

An invocation example to migrate existing partnership from IPv4 to IPv6 type

```
chpartnership -stop remote-sys-2
chpartnership -type ipv6 –clusterip fe80::200:f8ff:fe21:67cf remote-sys-2
```

The resulting output:

```
No feedback
```

An invocation example to configure a new CHAP secret for a partner

```
chpartnership –stop remote-system-2
chpartnership –chapsecret newpassword remote-system-2
chpartnership –start remote-system-2
```

The resulting output:

```
No feedback
```

An invocation example to configure a new system IP

```
chpartnership -stop remote-system-2
chpartnership –clusterip 202.49.86.2 –chapsecret newpassword remote-system-2
chpartnership –start remote-system-2
```

The resulting output:

```
No feedback
```

An invocation example setting the aggregate bandwidth and background copy rate

```
chpartnership -linkbandwidthmbits 2048 -backgroundcopyrate 100 localCluster
```

The resulting output:

```
No feedback
```

An invocation example enabling compression on an IP replication link

```
chpartnership -compressed yes svtcluster1
```

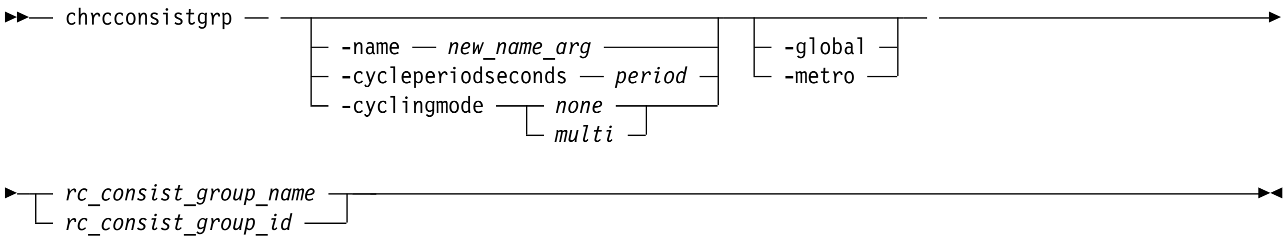
The resulting output:

```
No feedback
```

chrcconsistgrp

Use the **chrcconsistgrp** command to modify attributes of an existing Metro Mirror, Global Mirror, or `active-active` consistency group, such as changing the name of a consistency group.

Syntax



Parameters

-name *new_name_arg*
(Optional) Specifies the new name to assign to the consistency group.

-cycleperiodseconds *period*
(Optional) Specifies the cycle period in seconds. The minimum cycle period value is 60 seconds, and the default is 300 seconds.

This defines an optional cycle period that applies to Global Mirror relationships with a cycling mode of *multi*. A Global Mirror relationship that uses the *multi* `cycling_mode` performs a complete cycle each period. It might be provided for any relationship, but cannot be used for *none* in Metro or Global Mirror relationships.

-cyclingmode *none* | *multi*
(Optional) Specifies the behavior of Global Mirror for this relationship.

- Specifying *none*, the default, gives identical behavior to Global Mirror in previous versions of SAN Volume Controller .
- Specifying *multi* uses the cycling protocol.

To start a relationship with cycling_mode set to *multi*, change volumes must be defined for the relationship.

Note: The `cycling_mode` can be changed only when the relationship is stopped and in `consistent_stopped` or `inconsistent_stopped` states.

-metro
(Optional) Specifies the change in the consistency group's copy type and converts a Global Mirror (with or without change volumes) relationship to a Metro Mirror relationship.

Remember: To use this parameter the consistency group must be stopped (`inconsistent_stopped`, `consistent_stopped`, or `idling`)

-global
(Optional) Specifies the change in the consistency group's copy type and converts a Metro Mirror relationship to a Global Mirror relationship. This parameter is not mutually exclusive with **-cyclingmode**. If you do not specify **-cyclingmode** and the relationship is Metro Mirror, the `cycling_mode` value is `none`.

Remember: To use this parameter the consistency group must be stopped (`inconsistent_stopped`, `consistent_stopped`, or `idling`)

rc_consist_group_name | *rc_consist_group_id*
(Required) Specifies the ID or existing name of the consistency group that you want to modify.

Description

This command modifies the specified attributes of the supplied consistency group, one attribute at a time.

All parameters are mutually exclusive except for the **-cyclingmode**, which is mutually exclusive with all parameters but **-global**.

Note: One of the optional parameters must be specified.

You can change a relationship or consistency group between copy types even if replication is stopped. Consistency protection is preserved across all types, so a relationship or consistency group that is in `consistent_copying` state before it is stopped retains the consistent copy on the secondary system when the copying type is changed.

Note: You cannot set cycling mode to `multi-cycling` mode if there is a relationship where the primary and secondary volumes are different sizes.

A Global Mirror consistency group with cycling mode set to *multi* requires that change volumes are defined for the primary and secondary volumes of each relationship in the group before it can be started.

For intersystem relationships the **-cycleperiodseconds** and **-cyclingmode** parameters can be specified only when the two systems are connected. If the two systems become disconnected while the command is being processed, the command might complete with the change that is performed on the system that received the task invocation only. The other system is updated upon reconnection.

For consistency groups that are `active-active`, you cannot change the copy type or cycling mode. This means you cannot specify these parameters:

- **-global**

- **-metro**
- **-cyclingmode**

An invocation example to change a consistency group name from `rc_testgrp` to `rctestone`.

```
chrcconsistgrp -name rctestone rc_testgrp
```

The resulting output:

```
No feedback
```

chrcrelationship

Use the **`chrcrelationship`** command to modify certain attributes of an existing relationship, such as to add a relationship to a consistency group to remove a relationship from a consistency group, and to change the name of the relationship. You can change one attribute at a time.

Syntax

```
►►─ chrcrelationship ─ ┬── -masterchange ──┬─────────────────────────┬─┬─┬──────────┬──────────►
                       │                   ├─ master_change_vdisk_id ──┤ │ ├─ -global ─┤
                       │                   └─ master_change_vdisk_name ┘ │ └─ -metro ──┘
                       ├── -auxchange ──┬──────────────────────┬─────────┤
                       │                ├─ aux_change_vdisk_id ──┤        │
                       │                └─ aux_change_vdisk_name ┘        │
                       ├─ -nomasterchange ────────────────────────────────┤
                       ├─ -noauxchange ───────────────────────────────────┤
                       ├─ -name ─ new_name_arg ───────────────────────────┤
                       ├── -consistgrp ──┬─ consist_group_id ───┬─────────┤
                       │                 └─ consist_group_name ─┘         │
                       ├─ -noconsistgrp ──────────────────────────────────┤
                       ├─ -cycleperiodseconds ─ period ───────────────────┤
                       └── -cyclingmode ──┬─ none ──┬─────────────────────┘
                                          └─ multi ─┘

►─┬─ rc_rel_id ───┬──────────────────────────────────────────────────────────────────►◄
  └─ rc_rel_name ─┘
```

Parameters

-masterchange *master_change_vdisk_id* | *master_change_vdisk_name*
(Optional) Specifies a change volume association for the master volume in the relationship.

-auxchange *aux_change_vdisk_id* | *aux_change_vdisk_name*
(Optional) Specifies a change volume association for the auxiliary volume in the relationship.

-nomasterchange
(Optional) Specifies a defined change volume on the master volume must be removed from the relationship.

Note: To use this parameter the specified change volume must no longer be in use by the relationship, including change volumes of a running relationship (`inconsistent_copying`, `consistent_copying`, or `consistent_synchronized`).

This does not include a primary change volume of a stopped relationship. A secondary change volume of a relationship stopped from `consistent_copying` is considered in use if the change volume is supplying the consistent image. If this change volume needs to be removed, the relationships must first be stopped by specifying `stoprcrelationship -access` to apply the consistent image to the secondary volume.

-noauxchange
(Optional) Specifies a defined change volume on the auxiliary volume must be removed from the relationship.

Note: To use this parameter the specified change volume must no longer be in use by the relationship, including change volumes of a running relationship (`inconsistent_copying`, `consistent_copying`, or `consistent_synchronized`).

This does not include a primary change volume of a stopped relationship. A secondary change volume of a relationship stopped from `consistent_copying` is considered in use if the change volume is supplying the consistent image. If this change volume needs to be removed, the relationships must first be stopped by specifying `stoprcrelationship -access` in order to apply the consistent image to the secondary volume.

-name *new_name_arg*
(Optional) Specifies a new label to assign to the relationship.

-consistgrp *consist_group_id* | *consist_group_name*
(Optional) Specifies a new consistency group to assign the relationship to. Only relationships of the same copy type (Global Mirror, Metro Mirror, or `active-active`) can be assigned to the same consistency group.

-noconsistgrp
(Optional) Removes the specified relationship from a consistency group, making the relationship a standalone relationship.

-cycleperiodseconds *period*
(Optional) Specifies the cycle period in seconds. The minimum cycle period value is 60 seconds. The default is 300 seconds (5 minutes).

This defines an optional cycle period that applies to Global Mirror relationships with a cycling mode of multi. A Global Mirror relationship that uses the *multi* `cycling_mode` performs a complete cycle at most once each period.

-cyclingmode *none* | *multi*
(Optional) Specifies the behavior of Global Mirror for this relationship.

- Specifying *none*, the default, gives identical behavior to Global Mirror in previous versions of SAN Volume Controller .
- Specifying *multi* uses the cycling protocol.

To start a relationship with `cycling_mode` set to *multi*, change volumes must be defined for the relationship.

Note: The `cycling_mode` can be changed only when the relationship is stopped and in `consistent_stopped` or `inconsistent_stopped` status.

-metro
(Optional) Specifies the change in the relationship's copy type and converts a Global Mirror (with or without change volumes) relationship to a Metro Mirror relationship.

Remember: To use this parameter that the relationship must be stopped (`inconsistent_stopped`, `consistent_stopped`, or `idling`)

-global
(Optional) Specifies the change in the relationship's copy type and converts a Metro Mirror

relationship to a Global Mirror relationship. This parameter is not mutually exclusive with **-cyclingmode**. If you do not specify **-cyclingmode** and the relationship is Metro Mirror, the `cycling_mode` value is `none`.

Remember: To use this parameter that the relationship must be stopped (`inconsistent_stopped`, `consistent_stopped`, or `idling`)

rc_rel_name | *rc_rel_id*
(Required) Specifies the ID or name of the relationship.

Description

This command modifies the specified attributes of the supplied relationship, one attribute at a time. In addition to changing the name of a consistency group, this command can be used for the following purposes.

Remember:

- All parameters are mutually exclusive except for the **-cyclingmode**, which is mutually exclusive with all parameters but **-global**.
- One of the optional parameters must be specified.

Note: You cannot set cycling mode to `multi-cycling` mode if the primary and secondary volumes are different sizes.

- You can add a stand-alone relationship to a consistency group by specifying the **-consistgrp** parameter and the name or ID of the consistency group. The relationship and consistency group must be connected when the command is issued and must share the following components:
  - Master system
  - Auxiliary system
  - State (unless the group is empty)
  - Primary (unless the group is empty)
  - Type (unless the group is empty)
  - Cycling mode (unless the group is empty)

  When the first relationship is added to an empty group, the group takes on the same state, primary (copy direction), type (Metro Mirror or Global Mirror), and cycling mode as the relationship. Subsequent relationships must have the same state, copy direction, and type as the group in order to be added to it. A relationship can belong to only one consistency group.
- You can remove a relationship from a consistency group by specifying the **-noconsistgrp** parameter and the name or ID of the relationship. Although you do not have to specify or confirm the name of the consistency group, verify which group the relationship belongs to before you issue this command.

  This form of the modify relationship command succeeds in the connected or disconnected states. If the systems are disconnected the relationship is only removed from the consistency group on the local system, at the time the command is issued. When the systems are reconnected, the relationship is automatically removed from the consistency group on the other system. Alternatively, you can issue an explicit modify (**chrcrelationship**) command to remove the relationship from the group on the other system while it is still disconnected.

  Note: If you remove all relationships from the group, the relationship type is reset to `empty_group`. When you add a relationship to the empty group, the group again takes on the same type as the relationship.
- To move a relationship between two consistency groups, you must issue the **chrcrelationship** command twice. Use the **-noconsistgrp** parameter to remove the relationship from its current group, and then use the **-consistgrp** parameter with the name of the new consistency group.

You can change a relationship or consistency group between copy types even if replication is stopped. Consistency protection is preserved across all types, so a relationship or consistency group that is in `consistent_copying` state before it is stopped retains the consistent copy on the secondary system when the copying type is changed.

If you add a `consistent_stopped` relationship that uses consistency protection to a `consistent_stopped` consistency group that is not using consistency protection, the system attempts to trigger consistency protection on the consistency group that is not currently using consistency protection. If the relationship or consistency group that is not currently using consistency protection does not have secondary change volumes that are defined, specifying `chrcrelationship -consistgrp` fails. If you add a relationship to a consistency group where at least one is using consistency protection, the resulting consistency group is not mutually consistent, which means that the data on the secondary system for the relationship being added is inconsistent with the data on the consistency group. This also means that enabling access to the volume fails.

For intersystem relationships

- The **-name**, **-consistgrp**, **-cycleperiodseconds** and **-cyclingmode** parameters can only be specified when the two systems are connected. If the two systems become disconnected while the command is being processed, the command might be completed with the change performed on the system that received the task invocation only (and the other system is updated upon re-connection). The **-cycleperiodseconds** and **-cyclingmode** parameters can be specified only on stand-alone relationships (not members of a consistency group).
- The **-masterchange** and **-nomasterchange** parameters can only be specified when you run the **chrcrelationship** command on the master system for the relationship, and the **-auxchange** and **-noauxchange** parameters can only be specified when you run the **chrcrelationship** command on the auxiliary system for the relationship.

Remember: You cannot specify a master and auxiliary change volume in the same command.

A change volume must be:

- Used by the relationship that owns it
- In the same I/O group as the associated master or auxiliary volume
- The same size as the associated master or auxiliary volume

A change volume is owned and used by the associated Remote Copy relationship. Therefore, it cannot be:

- Mapped to a host
- Used as source or target of any FlashCopy maps
- Part of any other relationship
- A filesystem disk

Assigning a change volume to a relationship requires new FlashCopy mappings to be created between the master or auxiliary volume and the associated change volume. Therefore, there must be sufficient unallocated FlashCopy memory in the target I/O group or the command fails.

Note: You cannot use this command if cloud snapshot is enabled on the volume or the volume owner type is `cloud_backup`.

If the `cycle_period_seconds` for the relationship does not match that of the consistency group it is added to, the newly added relationship copies the `cycle_period_seconds` value from the group. If later removed from the group, the copied `cycle_period_seconds` value remains.

When a Global Mirror relationship with a `cycling_mode` value of *multi* is added to a group that is not empty, both the group and the relationship must be stopped.

For relationships that are `active-active`, you cannot change the copy type, cycling mode, or the change volumes. This means you cannot specify these parameters:

- **`-global`**
- **`-metro`**
- **`-cyclingmode`**
- **`-nomasterchange`**
- **`-noauxchange`**

Remember: For a volume to be configured as a change volume for an `active-active` relationship, the volume must have the same site name and site ID as the master and auxiliary volume it is being associated with.

An invocation example to change a relationship name from `rccopy1` to `testrel`

```
chrcrelationship -name testrel rccopy1
```

The resulting output:

```
No feedback
```

An invocation example to add relationship `rccopy2` to group `newgroup`

```
chrcrelationship -consistgrp newgroup rccopy2
```

The resulting output:

```
No feedback
```

An invocation example to remove relationship `rccopy3` from whichever consistency group it is a member of

```
chrcrelationship -noconsistgrp rccopy3
```

The resulting output:

```
No feedback
```

An invocation example

```
chrcrelationship -cyclingmode multi relB
```

The resulting output:

```
No feedback
```

An invocation example

```
chrcrelationship -cycleperiodseconds 20 relC
```

The resulting output:

```
No feedback
```

lspartnership

Use the **`lspartnership`** command to display a concise or detailed view of the current clustered systems (systems) that are associated with the local system.

Syntax

```
►►── lspartnership ── ┬─────────────────────────────────┬─ ┬────────────────────────┬─ ──────────────►
                      └─-filtervalue──attribute=value───┘  └─ -delim ── delimiter ──┘
```

```
►─┬──────────┬─ ┬─────────────────┬─ ┬───────────────┬──────────────────────►◄
  └─ -nohdr ─┘  └─-filtervalue?───┘  ├─system_id─────┤
                                     └─system_name───┘
```

Parameters

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed. If a capacity is specified, the units must also be included.

- Some filters allow the asterisk character (*) when you enter the command. The following rules apply to the use of wildcard characters with the SAN Volume Controller command-line interface (CLI):
  - The wildcard character is an asterisk (*).
  - The command can contain a maximum of one wildcard.
  - When you use a wildcard, you must enclose the filter entry within double quotation marks (""):

    ```
    lspartnership -filtervalue "name=md*"
    ```

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-filtervalue?
(Optional) displays a list of filters that can be applied against this view. The following filter attributes are valid:

- id
- name

system_id | *system_name*
(Optional) Specifies the name or ID of a system. Using this parameter displays the detailed view of the specific partner system, and any value that is specified by the **-filtervalue** (which filters a view that is based on specific attribute values that relate to each object type) parameter is ignored. When you specify system_id or system_name parameter, the concise view of all systems that match the filtering requirements that are specified by the **-filtervalue** parameter are displayed.

Description

Table 90 described attribute values.

Table 90. lspartnership attribute values

Attribute	Value
id	Indicates the system ID.
name	Indicates the system name.
location	Indicates the system location.

Table 90. lspartnership attribute values (continued)

Attribute	**Value**
code_level	Indicates the code level.
partnership	Indicates the current state of the partnership; not applicable for the local system and is blank. The partnership field can show the following values: **fully_configured** The `mkfcpartnership` or `mkippartnership` command is issued in both directions and the remote system is online and available. **partially_configured_local** The `mkfcpartnership` or `mkippartnership` command is issued from the local system to the remote system. The remote system is online and available for partnership. **partially_configured_local_stopped** The `mkfcpartnership` or `mkippartnership` command is issued from the local system to the remote system. The `chpartnership` command with the stop parameter is issued from the local system, and the remote system is online and available. Issue `chpartnership -start` on the local system, and `mkfcpartnership` or `mkippartnership` on the remote system **not_present** The `mkfcpartnership` or `mkippartnership` command is issued from the local system to the remote system, and the remote system is not available. The remote system is either offline or not connected to the local system. **fully_configured_stopped** The `mkfcpartnership` or `mkippartnership` command is issued in both directions and the remote system is online and available. The `chpartnership` command with the stop parameter is issued from the local system. **fully_configured_remote_stopped** The `mkfcpartnership` or `mkippartnership` command is issued in both directions and the remote system is online and available. The `chpartnership` command with the stop parameter is issued from the remote system. **fully_configured_local_excluded** The `mkfcpartnership` or `mkippartnership` command is issued in both directions. The local system excludes the connection to the remote system and the partnership is unable to sustain the I/O workload for the Metro Mirror or Global Mirror relationships. **fully_configured_remote_excluded** The `mkfcpartnership` or `mkippartnership` command is issued in both directions. The local system excludes the connection to the remote system and the partnership is unable to sustain the I/O workload for the Metro Mirror or Global Mirror relationships. **fully_configured_exceeded** There are too many systems in the system network and the partnership from the local system to the remote is disabled.
relationship_bandwidth_limit	Indicates the current bandwidth limit. **Important:** For partnerships over IP links with compression, this parameter specifies the aggregate bandwidth after the compression had been applied to the data. Do not set this parameter higher than the physical link bandwidth multiplied by the (carefully rounded down) compression factor.

Table 90. lspartnership attribute values (continued)

Attribute	Value
type	Indicates the type of partnership: • Fibre Channel (FC) • Internet Protocol Version 4 (IPv4) or Internet Protocol Version 6 (IPv6) FC partnerships are created connecting two systems over FC or Fibre Channel over Ethernet (FCoE) fabrics. IPv4 or IPv6 partnerships are created connecting two systems over native IP links.
cluster_ip	Indicates the partner system IP address, which can be IPv4 or IPv6. This information is displayed for IP-based partnerships. For IP-based partnership this field displays the system IP address that is specified while the partnership using **`mkippartnership`** is created.
chap_secret	Indicates the Challenge-Handshake Access Protocol (CHAP) secret (up to 80 alphanumeric characters) for the partner system. The CHAP authenticates the local system with the partner system during discovery, and Internet Small Computer System Interface (iSCSI) system session creation. For FC-based and FCoE-based relationships this field is always blank.
link_bandwidth_mbits	Indicates the aggregate bandwidth for the Remote Copy (RC) link in megabits per second (Mbps). This is a numeric value from `0` to `100000`. If there are multiple links between the local and remote systems, this parameter is set to the sum of the link bandwidths for these links.
background_copy_rate	Indicates the bandwidth allocation for background copy operations that are performed over the replication link. It is expressed as a percentage of the link bandwidth value, and is the maximum rate at which background copy operations are performed. This value is a number from `0` to `100` .
event_log_sequence	Indicates the last sequence number (indicating the last event) from event log for this partnership. This is a numeric value in the range 100 - 8000000. For FC-based and FCoE-based relationships this field is always blank.
max_replication_delay	Indicates the value for maximum replication delay. This value is a number in the range 100 - 360.
compressed	Indicates whether compression is enabled. The values is `yes` or `no` (default).

A concise invocation example

```
lspartnership
```

The concise resulting output:

```
id                name      location  partnership                type  cluster_ip                   event_log_sequence
000002006BC0A0D4  system-1  local
000002006200A0E5  system-2  remote    partially_configured_local ipv6  fe80::200:f8ff:fe21:67cf
000002006200A0F6  system-3  remote    partially_configured_local fc
000002006200A0G7  system-4  remote    partially_configured_local fc
```

A detailed invocation example

```
lspartnership cluster-2
```

The detailed resulting output:

```
id 000002006200A0EA
name system-2
location remote
partnership partially_configured_local
code_level 6.3.0.0 (build 35.7.1105071000)
console_IP 9.180.28.63:443
gm_link_tolerance 300
gm_inter_system_delay_simulation 0
```

```
gm_intra_system_delay_simulation 0
relationship_bandwidth_limit 25
gm_max_host_delay 5
type fc
cluster_ip
chap_secret
event_log_sequence
link_bandwidth_mbits 1024
background_copy_rate 25
max_replication_delay 145
compressed yes
```

lspartnershipcandidate

Use the **lspartnershipcandidate** command to list the clustered systems available for setting up a partnership with the local system. This is a prerequisite for creating inter-system Metro or Global Mirror relationships.

Syntax

```
►►— lspartnershipcandidate — ┬─────────┬─ ┬──────────────────────┬──────────────────────►◄
                             └ -nohdr ─┘  └ -delim — delimiter ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim : on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

Description

This command displays a list of systems that are available as candidate partner systems to form a Metro Mirror or Global Mirror partnership between two systems.

Output from the command shows the system ID, name, and configured status of the remote candidate system. The remote candidate system forms a partnership with the local system when you use the **mkippartnership** or **mkfcpartnership** command. The remote system shows the partnership status as partially_configured_local_stopped or partially_configured_local when you use the **lssystem** command. The **lspartnershipcandidate** command displays the configured status of those remote systems that form a partnership with the local system.

An invocation example

```
lspartnershipcandidate
```

The resulting output:

```
id               configured     system_name
0000010034E0F430 no             ldsystem26
```

lsrcconsistgrp

Use the **lsrcconsistgrp** command to return a concise list or a detailed view of remote copy relationships such as Metro Mirror, Global Mirror, or `active-active` consistency groups visible to the system.

Syntax

```
►►── lsrcconsistgrp ── ┬──────────────────────────────────────┬─ ┬─────────┬─ ───────────────────────►
                       └─ -filtervalue ── attribute=value ─┘   └─ -nohdr ┘

►┬──────────────────────┬─ ─ ┬────────────────┬┬─────────────┬──────────────────────────────────►◄
 └─ -delim ── delimiter ┘    └─ -filtervalue? ┘├─ object_id ──┤
                                               └─ object_name ┘
```

Parameters

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed. If a capacity is specified, the units must also be included.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards with the SAN Volume Controller command line interface (CLI):

- The wildcard character is an asterisk (*).
- The command can contain a maximum of one wildcard, which must be the first or last character in the string.
- When you use a wildcard, you must enclose the filter entry with double quotation marks (""), as follows:

```
lsrcconsistgrp -filtervalue "name=md*"
```

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

object_id | *object_name*
(Optional) Specifies the name or ID of an object. When you use this parameter, the detailed view of the specific object is displayed and any value that is specified by the **-filtervalue** parameter is ignored. If you do not specify the *object_id* | *object_name* parameter, the concise view of all objects that match the filtering requirements that are specified by the **-filtervalue** parameter are displayed.

-filtervalue?
(Optional) Specifies that you want your report to display any or all of the list of valid filter attributes. The following filter attributes for the **lsrcconsistgrp** command are valid:

- group_id
- name
- master_cluster_id
- master_cluster_name
- aux_cluster_id
- aux_cluster_name
- primary
- state
- relationship_count
- id
- copy_type

Description

This command returns a concise list or a detailed view of remote copy relationships such as Global Mirror, Metro Mirror, or `active-active` consistency groups that are visible to the system.

Table 91 provides possible values for the attributes that are displayed as data in the output views.

Table 91. lsrcconsistgrp command output values

Attribute	Value
primary	Indicates a primary consistency group. The values are `master` and `aux`.
state	Indicates the state. The values are: • `consistent_copying` • `inconsistent_stopped` • `inconsistent_copying` • `consistent_stopped` • `consistent_synchronized` • `idling` • `idling_disconnected` • `inconsistent_disconnected` • `consistent_disconnected` • empty
cycle_period_seconds	Indicates the minimum period in seconds between multiple cycles. The value is a number (integer) in a range in the range 60 - 86400. The default value is 300.
cycling_mode	Indicates the type of Global Mirroring, Metro Mirroring, or `active-active` cycling to use: `none` (default) or `multi`
freeze_time	Indicates the time in `YYMMDDHHMM` format

Table 91. lsrcconsistgrp command output values (continued)

Attribute	**Value**
status	Indicates the relationship status. The values are: • `online`, which indicates that the relationship is online and accessible. If the relationship state is `ConsistentSynchronized`, `ConsistentCopying`, or `InconsistentCopying` the volumes replicate host I/O operations that the primary volume receives. • `primary_offline`, which indicates that the primary volume in the relationship is offline. This prevents extra I/O operations, and synchronization pauses until the primary volume is online again. • `secondary_offline`, which indicates that the secondary volume in the relationship is offline. For r Global Mirror relationships in a `ConsistentSynchronized` state (no change volumes) and Metro Mirror relationships, extra I/O write operations to the primary volume terminate the relationship. • `io_channel_offline`, which indicates that the remote system is not accessible. For Global Mirror relationships in a `ConsistentSynchronized` state (no change volumes) and Metro Mirror relationships, extra I/O write operations to the primary volume terminate the relationship. • `primary_change_offline`, which indicates that the primary change volume in the relationship is offline. For Global Mirror with change volume relationships, the current I/O cycle ends, and a new I/O cycle begins when the primary change volume is online again. • `secondary_change_offline`, which indicates that the secondary change volume in the relationship is offline. For Global Mirror with change volume relationships, the current I/O cycle pauses and a new I/O cycle resumes when the secondary volume is online again. • `change_volumes_needed`, indicates an `active-active` relationship that is in a HyperSwap volume or Global Mirror volume with a change volume relationship. Additionally, at least one change volume is not configured. **Important:** Replication services are not usable. **Remember:** This field is blank.
sync	Indicates whether the consistency group is synchronized. The value is `in_syncor` `out_of_sync`.
mutually_consistent	Indicates whether the consistency group is mutually consistent. The values is `yes` or `no`. **Note:** This relationship is consistent with other consistency group relationships. This value is blank unless there is also a value in `consistent_stopped`, `consistent_disconnected`, and `consistent_copying`.
copy_type	Indicates the copy type. The values are: • `metro` • `global` • `activeactive` • blank

Note: The names of the Global Mirror or Metro Mirror relationships and consistency groups might be blank if the relationship or consistency groups are intersystem and the system partnership is disconnected.

The sync attribute has a value of `in_sync` when the contents are synchronized (identical) between volumes. If write operations take place on either the primary or secondary volume after a consistent (stopped) or idling state occurs, they will no longer be synchronized.

A concise invocation example

```
lsrcconsistgrp -delim :
```

The concise resulting output:

```
id:name:master_cluster_id:master_cluster_name:aux_cluster_id:aux_cluster_name:
primary:state:relationship_count:copy_type:cycling_mode:freeze_time

 248:jdemo_BA_cons1:0000020060406746:clusterB:0000020061413ABA:clusterA:master:
consistent_stopped:2:global:none:06/06/27/08/31/37
 249:rccstgrp0:0000020061413ABA:clusterA:0000020061413ABA:clusterA::empty:0
:empty_group
 250:jdemo_BA_cons2:0000020060406746:clusterB:0000020061413ABA:clusterA:master:
inconsistent_stopped:1:metro:none:06/06/27/08/31/37
 251:BA_cons1:0000020060406746:clusterB:0000020061413ABA:clusterA:master:
consistent_stopped:4:metro:none:06/06/27/08/31/37
 252:AB_cons2:0000020061413ABA:clusterA:0000020060406746:clusterB::empty:0
:empty_group:none:06/06/27/08/31/37
 253:AB_cons1:0000020061413ABA:clusterA:0000020060406746:clusterB:aux:
consistent_stopped:3:global:none:06/06/27/08/31/37
 254:AA_cons2:0000020061413ABA:clusterA:0000020061413ABA:clusterA::empty:0
:empty_group:none:06/06/27/08/31/37
 255:AA_cons1:0000020061413ABA:clusterA:0000020061413ABA:clusterA:master:
consistent_synchronized:2:global:none:06/06/27/08/31/37
```

A detailed invocation example

```
lsrcconsistgrp -delim : 254
```

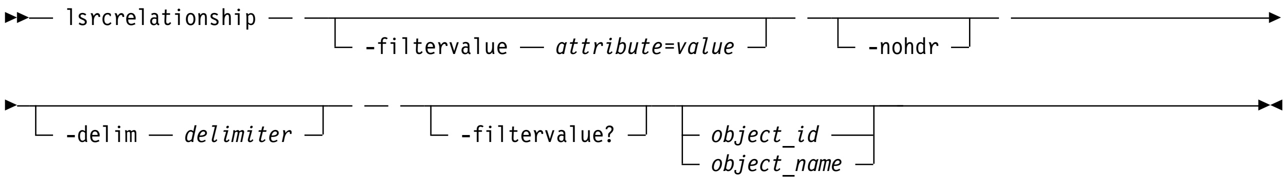
The detailed resulting output:

```
id:254
name:rccstgrp0
master_cluster_id:0000010030A007E5
master_cluster_name:clusterA
aux_cluster_id:0000010030A007E5
aux_cluster_name:clusterA
primary:master
state:consistent_synchronized
relationship_count:1
freeze_time:06/06/27/08/31/37
status:online
sync:in_sync
copy_type:activeactive
cycle_period_seconds:300
cycling_mode:none
RC_rel_id:2
RC_rel_name:aaa
```

lsrcrelationship

Use the **lsrcrelationship** command to return a concise list or a detailed view of remote copy relationships such as Metro Mirror, Global Mirror, or active-active relationships visible to the system.

Syntax



```
►►── lsrcrelationship ──┬──────────────────────────────────┬── ┬──────────┬── ──────────────────►
                        └─ -filtervalue ── attribute=value ┘   └─ -nohdr ─┘

►─┬─────────────────────┬── ── ┬─────────────────┬─┬─ object_id ───┬─────────────────────────────►◄
  └─ -delim ── delimiter┘      └─ -filtervalue? ─┘ └─ object_name ─┘
```

Parameters

-filtervalue *attribute=value*

(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are returned. If a capacity is specified, the units must also be included.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards with the SAN Volume Controller CLI:

- The wildcard character is an asterisk (*).
- The command can contain a maximum of one wildcard, which must be the first or last character in the string.
- When you use a wildcard, you must enclose the filter entry with double quotation marks (" "), as follows:

```
lsrcrelationship -filtervalue  "name=md*"
```

-nohdr

(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed even if the **-nohdr** parameter is specified.

-delim *delimiter*

(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

object_id | *object_name*

(Optional) Specifies the name or ID of an object. When you use this parameter, the detailed view of the specific object is returned and any value that is specified by the **-filtervalue** parameter is ignored. If you do not specify the *object_id* | *object_name* parameter, the concise view of all objects that match the filtering requirements that are specified by the **-filtervalue** parameter are displayed.

-filtervalue?

(Optional) Specifies that you want your report to display any or all of the list of valid filter attributes. The valid filter attributes for the **lsrcrelationship** command are:

- RC_rel_id
- RC_rel_name
- master_system_id
- master_system_name
- master_vdisk_id
- master_vdisk_name
- aus_system_id
- aux_system_name
- aux_vdisk_id
- aux_vdisk_name
- primary
- consistency_group_id
- consistency_group_name

- state
- progress
- copy_type

Description

This command returns a concise list or a detailed view of remote copy relationships such as Metro Mirror, Global Mirror, or `active-active` relationships visible to the system.

Table 92 provides possible values for the attributes that are displayed as data in the output views.

Table 92. **`lsrcrelationship`** *command attributes and values*

Attribute	**Value**
primary	Indicates a primary relationship. The values are `master` and `aux`.
state	Indicates the relationship state. The values are: • `consistent_copying` • `inconsistent_stopped` • `inconsistent_copying` • `consistent_stopped` • `consistent_synchronized` • `idling` • `idling_disconnected` • `inconsistent_disconnected` • `consistent_disconnected`
progress	Indicates the relationship progress. The value must be a number (integer) in the range 0 - 100.
cycle_period_seconds	Indicates the minimum period in seconds between multiple cycles. The value must be a number (integer) in the range 60 - 86400. The default is 300).
cycling_mode	Indicates the type of Global Mirror, Metro Mirroring, or `active-active` cycling to use. The values are `none` and `multi`.
copy_type	Indicates the copy type. The values are • `metro` • `global` • `activeactive` • blank
freeze time	Indicates the time in `YY/MM/DD/HH/MM` format

Table 92. **lsrcrelationship** *command attributes and values (continued)*

Attribute	Value
status	Indicates the status. The values are: • online indicates that the relationship is online and accessible. If the relationship state is ConsistentSynchronized, ConsistentCopying, or InconsistentCopying the volumes replicate host I/O operations that the primary volume receives. • primary_offline indicates that the primary volume in the relationship is offline. This prevents more I/O operations, and synchronization pauses until the primary volume is online again. • secondary_offline indicates that the secondary volume in the relationship is offline. For Global Mirror relationships in a ConsistentSynchronized state (no change volumes) and Metro Mirror relationships, extra I/O write operations to the primary volume terminate the relationship. • io_channel_offline indicates that the remote system is not accessible. For Global Mirror relationships in a ConsistentSynchronized state (no change volumes) and Metro Mirror relationships, extra I/O write operations to the primary volume terminate the relationship. • primary_change_offline indicates that the primary change volume in the relationship is offline. For Global Mirror with change volume relationships, the current I/O cycle ends, and a new I/O cycle begins when the primary change volume is online again. • secondary_change_offline indicates that the secondary change volume in the relationship is offline. For Global Mirror with change volume relationships, the current I/O cycle pauses and a new I/O cycle resumes when the secondary volume is online again. • change_volumes_needed, indicates an active-active relationship that is in a HyperSwap volumes or Global Mirror with change volume relationships. Additionally, at least one change volume is not configured. **Important:** Replication services are not usable.
sync	Indicates whether the relationship is synchronized. The value is in_sync or out_of_sync.
master_change_vdisk_name	Indicates the name of the volume that is acting as the master change volume for the relationship (blank if not defined) **Note:** This field identifies the change volume for the master volume if configured. For an intersystem relationship, if the master volume is in the other clustered system (system), the master change volume is also in the other system.
aux_change_vdisk_id	Indicates the ID of the volume that is acting as the auxiliary change volume for the relationship (blank if not defined) **Note:** This field identifies the change volume for the auxiliary volume, if such a volume is configured. For an intersystem relationship, if the auxiliary volume is in the other system, the auxiliary change volume is also in the other system.
aux_change_vdisk_name	Indicates the name of the volume that is acting as the auxiliary change volume for the relationship (blank if not defined) **Note:** This field identifies the change volume for the auxiliary volume if configured. For an intersystem relationship, if the auxiliary volume is in the other system, the auxiliary change volume is also in the other system.
bg_copy_priority	Unused.

Note: The names of the Global Mirror, Metro Mirror, or active-active relationships and consistency groups can be blank if the relationship or consistency groups are intersystem and the system partnership is disconnected.

The change_volumes_needed status is set if either the master or auxiliary change volume is not defined for relationships with either of these types:

- Copy type that is set to `active-active`
- Copy type that is set to `global` and `cycling_mode` set to `multi`

The sync attribute has a value of `in_sync` when the contents are synchronized (identical) between volumes. If write operations take place on either the primary or secondary volume after a consistent (stopped) or idling state occurs, they will no longer be synchronized.

A concise invocation example

```
lsrcrelationship -delim : -filtervalue name=j*
```

The concise resulting output:

```
id:name:master_cluster_id:master_cluster_name:master_vdisk_id:master_vdisk_name:
aux_cluster_id:aux_cluster_name:aux_vdisk_id:
aux_vdisk_name:primary:consistency_group_id:consistency_group_name:state:bg_copy
_priority:progress:copy_type:cycling_mode:freeze_time
45:jrel_AB1:0000020061413ABA:clusterA:45:jdisk_B8:0000020060406746:clusterB:38:j
disk_B1:master:::consistent_stopped:50:metro:none:06/06/27/08/31/37
48:jrel_AB2:0000020061413ABA:clusterA:48:jdisk_A4:0000020060406746:clusterB:41:j
disk_B4:master:::consistent_synchronized:50:metro:none:06/06/27/09/31/37
49:jrel_BA_1:0000020060406746:clusterB:42:jdisk_B5:0000020061413ABA:clusterA:49:j
disk_A5:master:248:jdemo_BA_cons1:consistent_stopped:50:metro:none:06/06/27/10/31/37
50:jrel_BA_2:0000020060406746:clusterB:43:jdisk_B6:0000020061413ABA:clusterA:
50:jdisk_A6:master:248:jdemo_BA_cons1:consistent_stopped:50:metro:none:06/06/27/11/31/37
```

A detailed invocation example

```
lsrcrelationship -delim : AB_2
```

The detailed resulting output:

```
id:9
name:AB_2
master_cluster_id:0000020061413MOE
master_cluster_name:chelseaB
master_vdisk_id:9
master_vdisk_name:stripe9
aux_cluster_id:0000020061413MOE
aux_cluster_name:chelseaB
aux_vdisk_id:10
aux_vdisk_name:stripe9_b
copy_type:activeactive
cycle_period_seconds:300
cycling_mode:multi
primary:master
consistency_group_id:
consistency_group_name:
state:consistent_synchronized
bg_copy_priority:50
progress:
freeze_time:2006/05/05/08/26/46
status:online
sync:in_sync
```

lsrcrelationshipcandidate

Use the **lsrcrelationshipcandidate** command to list volumes that can form Metro Mirror, Global Mirror, or `active-active` relationships. You can list eligible volumes that are on the local or remote clustered system (system).

Syntax

```
►►── lsrcrelationshipcandidate ──┬──────────────────────────────────────┬──────────────────────────────►
                                 └─ -master ──┬─ master_vdisk_id ───┬─┘
                                              └─ master_vdisk_name ─┘

►──┬─────────────────────────────────────┬── ──┬─────────┬── ──┬──────────────────────┬────────────────►◄
   └─ -aux ──┬─ aux_cluster_id ───┬──────┘     └─ -nohdr ─┘     └─ -delim ── delimiter ─┘
             └─ aux_cluster_name ─┘
```

Parameters

-master *master_vdisk_id* | *master_vdisk_name*
(Required) Specifies a particular volume to use as the master volume. The command finds candidates that match the size of this volume. If you are requesting candidate volumes on the local system, this command also matches the io_group.

-aux *aux_cluster_id* | *aux_cluster_name*
(Required) Specifies a remote system with volume candidates for an intersystem relationship. If you do not specify this parameter, the candidates on the local system are displayed.

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

Description

This command displays a list of volumes that can be either the master or the auxiliary disk for a Metro Mirror, Global Mirror, or `active-active` relationship. Volume IDs and names are displayed.

Note: Volumes that are flash disks are excluded from the view when a FlashCopy map is constructed.

An invocation example

```
lsrcrelationshipcandidate -delim :
```

The resulting output:

```
id:vdisk_name
0:vdisk0
4:vdisk4
```

lsrcrelationshipprogress

Use the **lsrcrelationshipprogress** command to display the progress of the background copy of a Metro Mirror, Global Mirror, or `active-active` relationship as a percentage. When the initial background copy process for a relationship completes, a null value is displayed for the progress of that relationship.

Syntax

```
►►── lsrcrelationshipprogress ── ┬─────────┬─ ┬──────────────────────┬─ ──────────────►
                                  └─ -nohdr ┘  └─ -delim ── delimiter ┘

►─┬─ rcrelationship_id ───┬────────────────────────────────────────────────────────►◄
  └─ rcrelationship_name ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (**:**) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

rcrelationship_id | *rcrelationship_name*
(Required) Specifies the object ID or name of the specified type.

Description

This command displays the progress of the background copy of a Metro Mirror, Global Mirror, or `active-active` relationship as a percentage.

An invocation example

```
lsrcrelationshipprogress -delim : 0
```

The resulting output:

```
id:progress
0:58
```

mkfcpartnership

Use the **mkfcpartnership** command to define partnerships using Fibre Channel (FC) or Fibre Channel over Ethernet (FCoE).

Syntax

```
►►──mkfcpartnership── ──-linkbandwidthmbits──link_bandwidth_in_mbps── ─────────────────────►

►─┬────────────────────────────────────┬─ ─┬─remote_system_id───┬──────────────────────►◄
  └─-backgroundcopyrate──percentage────┘   └─remote_system_name─┘
```

Parameters

-linkbandwidthmbits *link_bandwidth_in_mbps*

(Required) Specifies the aggregate bandwidth of the Remote Copy (RC) link between two clustered systems (systems in megabits per second (Mbps). It is a numeric value from 1 to 100000.

Note: Remote copy includes Metro Mirror and Global Mirror.
This value remains the same after a system update.

Note: If the specified value is non-zero, the combination of both the **-backgroundcopyrate** and the **-linkbandwidthmbits** values must result in a background copy bandwidth of at least 8 Mbps.

-backgroundcopyrate *percentage*

(Optional) Specifies the maximum percentage of aggregate link bandwidth that can be used for background copy operations. It is a numeric value from 0 to 100, and the default value is 50, which means that a maximum of 50% of the aggregate link bandwidth can be used for background copy operations. This parameter can be specified without stopping the partnership.

Note: If the specified value is non-zero, the combination of both the **-backgroundcopyrate** and the **-linkbandwidthmbits** values must result in a background copy bandwidth of at least 8 Mbps.

remote_system_id | *remote_system_name*

(Required) Specifies the remote system ID or name of a partnership. The specified value must match one of the system IDs or names returned after issuing **lspartnershipcandidate**.

Description

This command defines FC-based or FCoE-based partnerships. However, all existing partnerships are automatically updated to FC partnerships, any invocation of this command is applicable only to FC-based partnerships, and all partnerships created are FC-based partnerships.

An invocation example

```
mkfcpartnership –linkbandwidthmbits 100 –backgroundcopyrate 50 remote-system-2
```

The resulting output:

```
No feedback
```

An invocation example

```
mkfcpartnership –linkbandwidthmbits 1024 –backgroundcopyrate 25 remote-system-3
```

The resulting output:

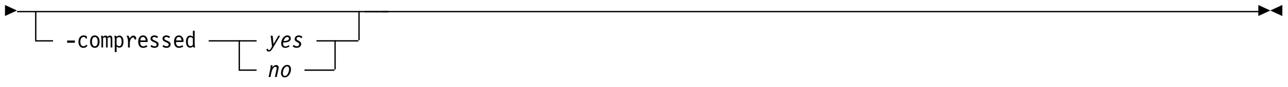
```
No feedback
```

mkippartnership

Use the **mkippartnership** command to define a new partnership created over Internet Protocol (IP) links.

Syntax

```
►►─mkippartnership── ─-type─┬─ipv4─┬─ ─┬──────────────────┬─ ─┬────────────────────────┬─ ──────►
                            └─ipv6─┘   └─-clusterip─ipadr─┘   └─-chapsecret─CHAPsecret─┘

►─-linkbandwidthmbits─link_bandwidth_in_mbps── ─┬──────────────────────────────┬─ ────────►
                                                └─-backgroundcopyrate─percentage─┘
```



Parameters

-type *ipv4* | *ipv6*

(Required) Specifies the Internet Protocol (IP) address format for the partnership using either of the following case-sensitive strings:

- *ipv4* for Internet Protocol Version 4 (IPv4)
- *ipv6* for Internet Protocol Version 6 (IPv6)

All Transmission Control Protocol (TCP) Remote Copy (RC) connections between the primary and remote clustered systems (systems) are created using specific IP addresses.

Note: Remote copy includes Metro Mirror and Global Mirror.
Partnership creation fails if the Internet Protocol (IP) address types specified for either primary or remote systems are not the same.

-clusterip *ipadr*

(Required) Specifies the partner system IP address, either *ipv4* or *ipv6*. Systems connected over IP links are not displayed by **lspartnershipcandidate** before executing **mkippartnership**. This does not apply to FC-based or FCoE-based connections.

-chapsecret *CHAPsecret*

(Optional) Specifies the Challenge-Handshake Authentication Protocol (CHAP) secret of the partner system. The maximum size of the CHAP secret is eighty alphanumeric characters.

-linkbandwidthmbits *link_bandwidth_in_mbps*

(Required) Specifies the aggregate bandwidth of the RC link between two clustered systems (systems) in megabits per second (Mbps). It is a numeric value from 1 to 100000.

Important: For partnerships over IP links with compression, this parameter specifies the aggregate bandwidth after the compression had been applied to the data. Do not set this parameter higher than the physical link bandwidth multiplied by the (carefully rounded down) compression factor.

This parameter can be specified without stopping the partnership.

Note: If the specified value is non-zero, the combination of both the **-backgroundcopyrate** and the **-linkbandwidthmbits** values must result in a background copy bandwidth of at least 8 Mbps.

-backgroundcopyrate *percentage*

(Optional) Specifies the maximum percentage of aggregate link bandwidth that can be used for background copy operations. It is a numeric value from 0 to 100, and the default value is 50, which means that a maximum of 50% of the aggregate link bandwidth can be used for background copy operations.

Note: If the specified value is non-zero, the combination of both the **-backgroundcopyrate** and the **-linkbandwidthmbits** values must result in a background copy bandwidth of at least 8 Mbps.

-compressed *yes* | *no*

(Optional) Specifies whether compression is enabled for this partnership. The default value is no.

Description

This command defines a new partnership created over Internet Protocol (IP) links. A remote system IP must be specified so its IP ports are enabled for data replication. RC sessions can then be created between the two partners.

In FC-based or FCoE-based partnerships, the partner system must first be a partnership candidate (displayed by **lspartnership**). Then it can become part of a partnership, created specifying **mkfcpartnership** with the remote system ID or name.

The **lspartnershipcandidate** command displays partnership candidates.

For IP partnerships, specifying **mkippartnership** with the cluster IP address and CHAP secret of the partner creates the partnership.

All TCP connections are established using either IPv4 or IPv6, and it cannot be a mix of the two IP address types.

Both systems in a partnership must have at least one IP address from an identical replication group to establish RC partnerships. Replication groups are numeric values that specify the pools of local IP addresses that establish Remote Copy partnerships with pools of IP addresses configured on the partner system.

An invocation example

```
mkippartnership –type ipv4 –clusterip 192.168.32.19
                –chapsecret mychapsecret –linkbandwidthmbits 100 –backgroundcopyrate 50
```

The resulting output:

```
No feedback
```

An invocation example

```
mkippartnership –type ipv6 –clusterip fe80::200:f8ff:fe21:67cf
                –chapsecret mychapsecret –linkbandwidthmbits 1024 –backgroundcopyrate 25
```

The resulting output:

```
No feedback
```

An invocation example creating a compressed IP replication link

```
mkippartnership –type ipv4 –clusterip 192.168.32.19 –chapsecret mychapsecret –linkbandwidthmbits 100 –backgroundcopyrate 50 -comp
```

The resulting output:

```
No feedback
```

mkpartnership (Discontinued)

The **mkpartnership** command is deprecated. Use either the **mkfcpartnership** or **mkippartnership** command instead.

mkrcconsistgrp

Use the **mkrcconsistgrp** command to create a new, empty remote copy consistency group. If the **-cluster** parameter is not specified, the consistency group is created on the local clustered system (system) only.

Syntax

```
►►— mkrcconsistgrp — ┬──────────────────────┬ ┬──────────────────────────────────┬──────────────►◄
                     └─ -name — new_name ─┘ └─ -cluster ─┬─ cluster_id ───┬─┘
                                                          └─ cluster_name ─┘
```

Parameters

-name *new_name*
(Optional) Specifies a name for the new consistency group.

-cluster *cluster_id* | *cluster_name*
(Optional) Specifies the name or ID of the remote system. If **-cluster** is not specified, a consistency group is created only on the local system.

Description

This command creates a new consistency group. The ID of the new group is displayed after the command processes. The name must be unique across all consistency groups that are known to the systems within this consistency group. If the consistency group involves two system, the systems must be in communication throughout the create process.

The new consistency group does not contain any relationships and is in an empty state. You can add Metro Mirror, Global Mirror, or `active-active` relationships to the group by using the **chrcrelationship** command.

Remember: Names representing remote copy consistency groups relationships are restricted to 15 characters in length (not sixty-three for an extended character set).

An invocation example

```
mkrcconsistgrp -name rc_testgrp
```

The resulting output:

```
RC Consistency Group, id [255], successfully created
```

mkrcrelationship

Specify the **mkrcrelationship** command to create a new Global Mirror, Metro Mirror, or `active-active` relationship with volumes in the same clustered system (system), forming an intrasystem Metro Mirror relationship or intersystem relationship (if it involves more than one system).

Syntax

```
►►── mkrcrelationship ── ── -master ──┬─ master_vdisk_id ───┬── ── -aux ──┬─ aux_vdisk_id ───┬── ──────►
                                      └─ master_vdisk_name ─┘              └─ aux_vdisk_name ─┘

►─ -cluster ──┬─ cluster_id ───┬── ──┬─────────────────────────┬── ──────────────────────────────►
              └─ cluster_name ─┘     └─ -name ── new_name_id ──┘

►──┬───────────────────────────────────────────────┬── ──┬──────────┬── ────────────────────────►
   └─ -consistgrp ──┬─ consist_group_id ───┬───────┘     └─ -sync ──┘
                    └─ consist_group_name ─┘

►──┬──────────────────────────────────────────────┬── ──────────────────────────────────────────►◄
   ├─ -global ──┬──────────────────────────────┬──┤
   │            └─ -cyclingmode──┬─ none ──┬───┘  │
   │                             └─ multi ─┘      │
   └─ -activeactive ──────────────────────────────┘
```

Parameters

-master *master_vdisk_id* | *master_vdisk_name*
(Required) Specifies the ID or name of the *master_vdisk_id* or *master_vdisk_name.*

If a new remote copy relationship is mapped to a host of type hide_secondary, the secondary volume is not presented to the host; however, it is mapped for configuration purposes. The secondary volume is presented to the host if the:

- Host type is changed to a type other than hide_secondary
- Remote copy relationship is stopped by specifying stoprcrelationship -access
- Volume is no longer a secondary volume because the remote copy relationship is deleted or switched

-aux *aux_vdisk_id* | *aux_vdisk_name*
(Required) Specifies the ID or name of the aux_vdisk_id or aux_vdisk_name.

-cluster *cluster_id* | *cluster_name*
(Required) Specifies the ID or name of the remote cluster.

- If you are creating an intrasystem relationship, enter the ID of the local system. The volumes in the relationship must belong to the same I/O group within the system.
- If you are creating an intersystem relationship, enter the ID of the remote system. To create a relationship in two different systems, the systems must be connected at the time that the **mkrcrelationship** command is received.

-name *new_name_id*
(Optional) Specifies a label to assign to the relationship.

-consistgrp *consist_group_id* | *consist_group_name*
(Optional) Specifies a consistency group that this relationship joins. If you do not supply the **-consistgrp** parameter, the relationship is created as a stand-alone relationship that can be started, stopped, and switched on its own.

Note: Metro Mirror, Global Mirror, and active-active relationships cannot belong to the same consistency group. When the first relationship is added to the consistency group, the group takes on the same type as the relationship. Then, only relationships of that type can be added to the consistency group.

-sync
(Optional) Specifies that you want the system to create a synchronized relationship. The **-sync** parameter guarantees that the master and auxiliary disks contain identical data at the point that the relationship is created. You must ensure that the auxiliary disk is created to match the master disk and that no input transactions take place to either disk before you issue the create command. The initial background synchronization is skipped.

-global
(Optional) Specifies that you want the system to create a new Global Mirror relationship. If you do not specify the **-global** parameter, a Metro Mirror relationship is created instead. You cannot specify this keyword with **-activeactive**.

-cyclingmode *none* | *multi*
(Optional) Specifies the behavior of Global Mirror for this relationship.

- Specifying *none*, the default, gives identical behavior to Global Mirror in previous versions of SAN Volume Controller .
- Specifying *multi* uses the cycling protocol.

The default cycle period is 300 seconds. The cycle period can be modified after the relationship is created by using the **chrcrelationship** command. To start a relationship with cycling_mode set to *multi*, change volumes must be defined for the relationship.

Important: This parameter must be specified with **-global**.

-activeactive
(Optional) Specifies that the relationship is created in an active-active mode. You cannot specify this keyword with **-global** (this parameter defaults to a Metro Mirror relationship that is being created).

Description

This command creates a new Global Mirror, Metro Mirror, or `active-active` relationship. A Metro Mirror relationship defines the relationship between two volumes. One volume is a master volume and the other volume is an auxiliary volume. This relationship persists until deleted. The auxiliary volume must be identical in size to the master volume or the command fails. This command also returns the new relationship ID.

The master and auxiliary cannot be in an existing relationship. Any defined FlashCopy mappings that have the proposed master volume as the target of the FlashCopy mapping must be using the same I/O group as the master volume. Any defined FlashCopy mappings that have the proposed auxiliary volume as the target of the FlashCopy mapping must be using the same I/O group as the auxiliary volume.

Note: You cannot create a remote copy relationship with this command if the auxiliary volume is an active FlashCopy mapping target. If the I/O group has enough bitmap space available to allocate for remote copy and the allocated space for the remote copy is not large enough to accommodate the new relationship, space is automatically added. (Remote copy includes Global Mirror, Metro Mirror, and `active-active` relationships.)

Note: You cannot use this command if cloud snapshot is enabled on the volume or the volume owner type is `cloud_backup`.

Metro Mirror relationships use one of the following copy types:

- A Metro Mirror copy ensures that updates are committed to both the primary and secondary volumes before the copy sends confirmation of I/O completion to the host application. This ensures that the secondary volume is synchronized with the primary volume if a failover operation is performed.
- A Global Mirror copy allows the host application to receive confirmation of I/O completion before the updates are committed to the secondary volume. If a failover operation is performed, the host application must recover and apply any updates that were not committed to the secondary volume.

You can optionally give the relationship a name. The name must be a unique relationship name across both systems.

The relationship can optionally be assigned to a consistency group. A consistency group ensures that a number of relationships are managed so if the relationships disconnect, the data in all relationships within the group is in a consistent state. For example, the state can be important in a database application where data files and log files are stored on separate volumes and consequently are managed by separate relationships.

Remember: In the event of a disaster, the primary and secondary sites might become disconnected. As the disconnection occurs and the relationships stop copying data from the primary to the secondary site, there is no assurance that updates to the two separate secondary volumes stop in a consistent manner if the relationships that are associated with the volumes do not belong to a consistency group.

For proper database operation, it is important that updates to the log files and the database data are made in a consistent and orderly fashion. It is crucial in this example that the log file volume and the data volume at the secondary site are in a consistent state. This can be achieved by putting the relationships that are associated with these volumes into a consistency group. Both Metro Mirror and Global Mirror processing ensure that updates to both volumes at the secondary are stopped, leaving a consistent image based on the updates that occurred at the primary site.

If you specify a consistency group, both the group and the relationship must are created by using the same master system and the same auxiliary system. The relationship must not be a part of another consistency group. If the consistency group is empty, it acquires the type of the first relationship that is added to it. Therefore, each subsequent relationship that you add to the consistency group must have the same type.

If the consistency group is not empty, the consistency group and the relationship must be in the same state. If the consistency group is empty, it acquires the state of the first relationship that is added to it. If the state has an assigned copy direction, the direction of the consistency group and the relationship must match that direction.

If you do not specify a consistency group, a stand-alone relationship is created.

If you specify the **-sync** parameter, the master and auxiliary volumes contain identical data at the point when the relationship is created. You must ensure that the auxiliary is created to match the master and that no data movement occurs to either volume before you issue the **mkrcrelationship** command.

If you specify the **-global** parameter, a Global Mirror relationship is created. Otherwise, a Metro Mirror relationship is created instead.

The volumes that are specified on the **-master** and **-aux** parameters cannot be master or auxiliary volumes in an existing relationship.

If you specify **-activeactive**:

- The system that is specified with **-cluster** must be the local system.
- **-global** must not be specified.
- The volume that is specified with **-master** must:
  - Be in an I/O group with both nodes that have the same site name and site ID
  - Have all volume copies stored in storage pools in the same site as the volume's I/O group
  - Not be the target of a FlashCopy mapping
  - Not be the source of any FlashCopy mappings to volumes in a different site or by using bitmap memory from nodes in a different site (but the volume can be the source of a FlashCopy mapping in which the target volume and map are in the same site)
- The volume that is specified with **-aux** must:
  - Be part of an I/O group with a different site ID and site name than the master volume (with no volume host mappings that are defined)
  - Have all volume copies stored in storage pools in the same site as the volume's I/O group
  - Not be the target of a FlashCopy mapping
  - Not be the source of any FlashCopy mappings to volumes in a different site or by using bitmap memory from nodes in a different site (but the volume can be the source of a FlashCopy mapping in which the target volume and map are in the same site)

Access the data stored on these volumes by accessing the volume you specify using the **-master** parameter. Both I/O groups of the volumes that are specified by the **-master** and **-aux** parameters have a local physical copy and cache, allowing access (by using the master volume ID) whether the auxiliary volume's site is available or not.

Remember: This command cannot be used on a volume that is owned by a file system.

An invocation example

```
mkrcrelationship -master vdisk1 -aux vdisk2 -name rccopy1
 -cluster 0000020063432AFD
```

The resulting output:

```
RC Relationship, id [28], successfully created
```

An invocation example

```
mkrcrelationship -master vdiskA -aux vdiskB -cluster clusterB -name new_rel -global -cyclingmode multi
```

The resulting output:

```
RC Relationship, id [28], successfully created
```

An invocation example

```
mkrcrelationship -master volA -aux volB -cluster localCluster -activeactive
```

The resulting output:

```
RC Relationship, id [28], successfully created
```

rmpartnership

Use the **rmpartnership** command to remove a Metro Mirror or Global Mirror partnership on one clustered system (system). Because the partnership exists on both systems, it is necessary to run this command on both systems to remove both sides of the partnership. If the command is run on only one system, the partnership enters a partially configured state on the other system.

Syntax

```
►►─ rmpartnership ─ ┬─ remote_cluster_id ───┬──────────────────────────────────────────►◄
                    └─ remote_cluster_name ─┘
```

Parameters

remote_cluster_id | *remote_cluster_name*
(Required) Specifies the system ID or the name of the remote system.

Description

This command deletes one half of a partnership on a system. To remove the entire partnership, you must run the command twice, once on each system.

Attention: Before running the **rmpartnership** command, you must remove all relationships and groups that are defined between the two systems. To display system relationships and groups, run the **lsrcrelationship** and **lsrcconsistgrp** commands. To remove the relationships and groups that are defined between the two systems, run the **rmrcrelationship** and **rmrcconsistgrp** commands.

An invocation example

```
rmpartnership cluster1
```

The resulting output:

```
No feedback
```

rmrcconsistgrp

Use the **rmrcconsistgrp** command to delete an existing Metro Mirror, Global Mirror, or `active-active` consistency group.

Syntax

```
►►── rmrcconsistgrp ── ─┬────────┬─ ─┬─ rc_consist_group_id ───┬──────────────────►◄
                        └─ -force ┘   └─ rc_consist_group_name ─┘
```

Parameters

-force
(Optional) Specifies that you want the system to remove any relationship that belongs to a group before the consistency group is deleted. The relationship itself is not deleted; it becomes a stand-alone relationship.

Note: The **-force** parameter must be used to delete a consistency group when the consistency group has any Metro Mirror, Global Mirror, or `active-active` relationship that is associated with it. If you do not use the **-force** parameter, the command fails.

rc_consist_group_id | *rc_consist_group_name*
(Required) Specifies the ID or the name of the consistency group to delete.

Description

This command deletes the specified consistency group. You can issue this command for any existing consistency group. If the consistency group is disconnected at the time that the command is issued, the consistency group is only deleted on the cluster that is connected. When the clusters reconnect, the consistency group is automatically deleted on the other cluster. Alternatively, if the clusters are disconnected, and you still want to remove the consistency group on both clusters, you can issue the **rmrcconsistgrp** command separately on both of the clusters.

If the consistency group is not empty, the **-force** parameter is required to delete the group. Specifying the **-force** parameter removes the relationships from the consistency group before the group is deleted. These relationships become stand-alone relationships. The state of these relationships is not changed by the action of removing them from the consistency group.

Note: Using the force parameter might result in a loss of access. Use it only under the direction of your product support information.

An invocation example

```
rmrcconsistgrp rctestone
```

The resulting output:

```
No feedback
```

rmrcrelationship

Use the **rmrcrelationship** command to delete an existing remote copy relationship.

Syntax

```
►►── rmrcrelationship ── ─┬────────┬─ ─┬─ rc_rel_id ───┬──────────────────────────►◄
                          └─ -force ┘   └─ rc_rel_name ─┘
```

Parameters

-force
(Optional) Specifies that the relationship must be deleted even if it results in the secondary volume that contains inconsistent data. This parameter applies only to `active-active` relationships or Global Mirror relationships that use multi cycling mode.

rc_rel_id | *rc_rel_name*
(Required) Specifies the ID or the name of the relationship.

Description

This command deletes the relationship that is specified. Deleting a relationship removes the logical relationship between the two volumes but it does not affect the volumes themselves.

If the relationship is disconnected at the time that the command is issued, the relationship is only deleted on the clustered system (system) where the command is being run. When the systems reconnect, the relationship is automatically deleted on the other system. Alternatively, if the systems are disconnected and if you still want to remove the relationship on both systems, you can issue the **rmrcrelationship** command independently on both of the systems.

If a relationship is `active-active` or is a Global Mirror relationship that uses multicycling mode, and you attempt to delete the relationship without enabling access first, specifying **rmrcrelationship** might fail with an error because the relationship does not currently have a fully consistent secondary volume. Specifying **-force** overrides this test. This is not the default behavior, and you can quiesce and delete the relationship in order to use the secondary volume's data immediately. If the map is still performing the background copy to migrate data from the change volume to the secondary volume, the changed volume and associated FlashCopy mappings remain defined when **rmrcrelationship** completes. The FlashCopy mappings are deleted after the background copy completes, and the change volume becomes unusable again.

If you delete an inconsistent relationship, the secondary volume becomes accessible even though it is still inconsistent. This case is the only one in which Metro Mirror, Global Mirror, or HyperSwap does not inhibit access to inconsistent data.

An invocation example

```
rmrcrelationship rccopy1
```

The resulting output:

```
No feedback
```

startrcconsistgrp

Specify **startrcconsistgrp** to start the Global Mirror, Metro Mirror, or `active-active` consistency group copy process, set the direction of copy if it is undefined, and optionally mark the secondary volumes of the consistency group as clean.

Syntax

```
►►── startrcconsistgrp ── ┬──────────────────────────┬─ ┬──────────┬─ ┬──────────┬─ ──────────►
                          └─ -primary ──┬─ master ─┬─┘ └─ -force ─┘ └─ -clean ─┘
                                        └─ aux ────┘

►──┬─ rc_consist_group_id ───┬──────────────────────────────────────────────────────────►◄
   └─ rc_consist_group_name ─┘
```

Parameters

-primary *master* | *aux*
(Optional) Specifies the copy direction by defining whether the master or auxiliary disk becomes the primary (source). This parameter is required when the primary is undefined if, for example, the consistency group is in the `Idling` state.

-force
(Optional) Specifies that you want the system to process the copy operation even if it might lead to a temporary loss of consistency while synchronization occurs. This parameter is required if the consistency group is in the `ConsistentStopped` state, but is not synchronized or is in the `idling` state - except if consistency protection is configured.

-clean
(Optional) Specifies that the volume indicated as secondary is clean for each of the relationships that belong to the group. Any changes that are made on the secondary volume are ignored, and only changes made on the clean primary volume are considered during synchronization of the primary and secondary disks. The consistency group must be in an Idling (connected) state for this parameter to work.

Attention: Specify this parameter when all data changed on the secondary volumes while the consistency group was in the `idling` state matches the state of the primary volumes when the consistency group was stopped. Otherwise, relationships that are not consistent are reported as consistent. Once this has been done, there is no method to determine whether these volumes ever reach a true consistent state until a full background copy can be carried out again.

rc_consist_group_id | *rc_consist_group_name*
(Required) Specifies the ID or name of the consistency group to start.

Description

This command starts a Global Mirror, Metro Mirror, or `active-active` stand-alone consistency group. You cannot use this command to start a remote copy relationship if the primary volume is a target volume of a prepared FlashCopy mapping.

This command can be issued only to a consistency group that is connected. For a consistency group that is idling, this command assigns a copy direction (primary and secondary roles) and begins the copy process. Otherwise, this command restarts a previous copy process that was stopped either by a stop command or by an I/O error.

Note: You cannot start a consistency group if it contains a relationship with primary and secondary volumes that are different sizes.

If you specify `stoprcconsistgrp -access` on an existing remote copy relationship that is restarted and the resultant secondary volume (depending on the choice of primary) is mapped to a host of type `hide_secondary`, that volume is not presented to the host. This is true even though it is mapped for configuration purposes. The mapped volumes are presented to the host if the:

- Host type is changed to a type other than `hide_secondary`
- Remote copy relationship is stopped and you specify `stoprcconsistgrp -access`
- Volume ceases to be a secondary volume because the remote copy relationship is being deleted or switched

If the resumption of the copy process leads to a period when the relationship is not consistent, then you must specify the **-force** parameter when you restart the relationship. This situation can arise if the relationship is stopped and then further input transactions is performed on the original primary disk of the relationship. When you use the **-force** parameter in this situation, the data on the secondary disk is not usable (because it is inconsistent) in a disaster recovery circumstance.

In the idling state, you must provide the **-primary** parameter. In other connected states, you can provide the **-primary** parameter, but it must match the existing setting.

The **-force** parameter is required if consistency would be lost by starting a copy operation. This can occur if write operations on either primary or secondary volumes take place since the `ConsistentStopped` or `idling` state occurred. If the command is issued without the **-force** parameter in such circumstances, the command fails . In general, the **-force** parameter is required if the group is in one of the following states:

- `consistent_stopped` but not synchronized (`sync=out_of_sync`)
- `i` but not synchronized (`sync=out_of_sync`)

The **-force** parameter is not required if the group is in `inconsistent_stopped`, `inconsistent_copying`, or `consistent_synchronized` state. The command does not fail if you specify the **-force** parameter.

When you configure a secondary change volume on all relationships in a consistency group, the consistency group changes to `consistent_copying` state during resynchronization. If you specify **startrcconsistgrp** for a consistency group that is in an `idling` state, consistency protection is disabled if any of the secondary volumes are written to. This means that you must specify the **-force** parameter.

The **-clean** parameter is used when a Global Mirror or Metro Mirror group is started and the secondary volumes in this group are assumed to be clean, which means that any changes that are made at the secondary are ignored and only changes that are made at the primary are considered when synchronizing the primary and secondary volumes. The **-clean** parameter can be used in the following scenario:

1. You specify the **-sync** parameter to create a consistency group. It does not matter if the primary and secondary contain the same data - even though the use of the **-sync** parameter implies that is true.
2. You specify `stoprcconsistgrp -access`. This permits access to the secondary disk. Change recording begins at the primary.
3. You copy and load an image of the primary disk onto to the secondary disk. It is permissible to allow updates to the primary disk during the image copy as this image can be only a fuzzy image of the primary disk.
4. You specify the **startrcconsistgrp** command with the **-primary master**, **-force**, and **-clean** parameters. The auxiliary disk is marked as clean and changes on the master disk that occur since the relationship was stopped are copied to the auxiliary disk.
5. A background copy completes and the relationships in the group become consistent and synchronized.

After you restart a consistency group in either of these states (`Idling` or `multi`), the data on the secondary volumes is not usable for disaster recovery until the consistency group becomes consistent.

A Global Mirror consistency group with a `cycling_mode` of `multi` in either of these states does not require the **-force** parameter because consistent secondary images are retained. However, if such a consistency group is in `idling` state and written data is received at any secondary volume in the consistency group, the **-force** flag is still required, because the secondary volumes have a divergent image that cannot represent a consistent earlier state.

A Global Mirror relationship with a cycling mode of:

- *none* uses the non-cycling Global Mirror algorithm
- *multi* must have a change volume that is configured at the primary volume (or the command fails)
- *multi* must also have a change volume that is configured at the secondary volume (or the command fails)
- *multi* performs multiple cycles of cycling

After you create a background copy the relationship remains in copying state, waiting for the remainder of the period time to expire before you perform a new cycle. If the secondary change volume is deconfigured when the background copy completes, the relationship stops as if there is no cycle period.

Relationships that are `active-active` must have a state of `idling` to be started.

An invocation example

```
startrcconsistgrp rccopy1
```

The resulting output:

```
No feedback
```

startrcrelationship

Use the **`startrcrelationship`** command to start the Metro Mirror or Global Mirror relationship copy process, set the direction of copy if undefined, and (optionally) mark the secondary volume of the relationship as clean. The relationship must be a stand-alone relationship. You can also use this command to restart the `active-active` relationship copy process after you specify `stoprcrelationship -access`.

Syntax

```
►►— startrcrelationship — ────────────────────────── ──────────── ──────────── ────────►
                           └ -primary ─┬─ master ─┬┘ └ -force ─┘  └ -clean ─┘
                                       └─ aux ────┘

►─┬─ rc_rel_id ───┬──────────────────────────────────────────────────────────────────►◄
  └─ rc_rel_name ─┘
```

Parameters

-primary *master* | *aux*
: (Optional) Specifies the copy direction by defining whether the master or auxiliary disk becomes the primary (source). This parameter is required when the primary is undefined if, for example, the relationship is in the idling state.

-force
: (Optional) Specifies that you want the system to process the copy operation even if it might lead to a temporary loss of consistency while synchronization occurs. This parameter is required if the relationship is in the `ConsistentStopped` state, but is not synchronized or in `idling` state - except if consistency protection is configured.

  Important: Using the force parameter might result in a loss of access. Use it only under the direction of your product support information.

-clean
: (Optional) Specifies that the volume that is to become a secondary is clean. Any changes that are made on the secondary volume are ignored, but changes made on the clean primary volume are considered when synchronizing the primary and secondary disks. The relationship must be in an Idling (connected) state for this parameter to work.

  Attention: This flag must be used only if all data changed on the secondary volumes while the consistency group was in the `idling` state matches the state of the primary volumes when the consistency group was stopped. Otherwise, relationships that are not consistent are reported as consistent. When this completes, there is no method to determine whether these volumes ever reach a true consistent state until a full background copy can be carried out again.

rc_rel_id | *rc_rel_name*
(Required) Specifies the ID or name of the relationship that you want to start in a stand-alone relationship.

Description

The **startrcrelationship** command starts a stand-alone relationship. The command fails if it is used to start a relationship that is part of a consistency group.

Note: You cannot start a relationship if the primary and secondary volumes are different sizes.

This command can be specified only to a relationship that is connected. For a relationship that is idling, this command assigns a copy direction (primary and secondary roles) and begins the copy process. Otherwise, this command restarts a previous copy process that was stopped either by a stop command or by some I/O error.

Note: A command in idling state is rejected if any of the indicated secondary volumes is the target of an existing FlashCopy map.

If the FlashCopy mapping is active, the remote copy cannot be started.

If an existing remote copy relationship is stopped by specifying `stoprcrelationship -access` but is restarted and the resultant secondary volume (depending on the choice of primary) is mapped to a host of type `hide_secondary`, that volume is not presented to the host. This is true even though it is mapped for configuration purposes. The mapped volumes are presented to the host if the:

- Host type is changed to a type other than `hide_secondary`
- Remote copy relationship is stopped by specifying `stoprcrelationship -access`
- Volume ceases to be a secondary volume because the remote copy relationship is being deleted or switched

In the idling state, you must provide the **-primary** parameter. In other connected states, you can provide the **-primary** parameter, but it must match the existing setting.

The **-force** parameter is required if consistency would be lost by starting a copy operation. This situation can occur if input transactions occur on either the primary or secondary volumes since the `ConsistentStopped` or `Idling` state occurred. This situation occurs when the relationship is in either of these states:

- `ConsistentStopped` but not synchronized
- `Idling` but not synchronized

After restarting a relationship in either of these states, the data on the secondary volume is not usable for disaster recovery until the relationship becomes consistent.

A Global Mirror relationship with a `cycling_mode` of `multi` in either of these states does not require the **-force** parameter because a consistent secondary image is retained. However, if such a relationship is in idling state and written data is received at the secondary volume, the **-force** flag is required because the secondary volume has a divergent image that cannot represent a consistent earlier state.

The **-force** parameter is not required if the relationship is in one of the following states:

- `InconsistentStopped`
- `InconsistentCopying`
- `ConsistentSynchronized`

However, the command does not fail if you specify the **-force** parameter.

You do not have to specify the **-force** parameter for relationships with configured secondary change volumes. If you specify **startrcrelationship** for an `idling` relationship, consistency protection is disabled if the secondary volume is written to. This means that you must specify the **-force** parameter.

A Global Mirror relationship with a cycling mode of:

- *none* uses the non-cycling Global Mirror algorithm
- *multi* must:
  - Use a change volume that is configured at the primary volume (or the command fails)
  - Use a change volume that is configured at the secondary volume (or the command fails)
  - Perform multiple cycles of cycling

After you create a background copy the relationship remains in copying state, wait for the remainder of the period time to expire before you perform a new cycle. If the secondary change volume is unconfigured when the background copy completes, the relationship stops as if there is no cycle period.

Relationships that are `active-active` must have a state of `idling` to be started. (You must specify **-primary** to determine which of the master and auxiliary copies become the primary when you start an `idling` relationship.)

Use this command to:

- Restart the `active-active` relationship copy process and retain the historical disaster recovery copy that access is granted to (which might be used while the up-to-date copy was offline)
- Switch back to an up-to-date copy in the same state it was in before you specify `stoprcrelationship -access`. Any changes that are made to the historical copy are discarded

Remember: If you switch back to the up-to-date copy, you might have to take host actions to prepare for the volume data that changes.

After you specify this command, if the secondary copy is not a historical copy of the primary relationship, it cannot be used for disaster recovery (and disaster recovery availability is restored after the copies are resynchronized). This situation can occur when:

- The new primary is the historical copy, which means the new secondary copy contains data that is from a later point in time than the data that the primary contains
- The secondary copy is the historical copy and is modified between specifying `stoprcrelationship -access` and `startrcrelationship -primary` command (which means the secondary copy represents a divergent data image)

This command copies only the regions that are needed to resynchronize the two copies.

An invocation example

```
startrcrelationship rccopy1
```

The resulting output:

```
No feedback
```

stoprcconsistgrp

Use the **stoprcconsistgrp** command to stop the copy process for a Metro Mirror, Global Mirror, or `active-active` consistency group. This command can also be used to enable write access to the secondary volumes in the group if the group is in a consistent state.

Syntax

```
►►── stoprcconsistgrp ── ─┬──────────┬─ ─┬─ rc_consist_group_id ───┬──────────────────────────►◄
                          └─ -access ─┘  └─ rc_consist_group_name ─┘
```

Parameters

-access
(Optional) Allows write access to consistent secondary volumes in the consistency group.

rc_consist_group_id | *rc_consist_group_name*
(Required) Specifies the ID or the name of the consistency group to stop all processing for.

Description

This command applies to a consistency group. You can issue this command to stop processing on a consistency group that is copying from primary volumes to secondary volumes.

Note: You cannot stop a consistency group by using the **-access** parameter for a relationship if the primary and secondary volume are different sizes.

If the consistency group is in an inconsistent state, all copy operations stop and do not resume until you issue the **startrcconsistgrp** command. When a consistency group is in a consistent state (consistent_stopped, consistent_synchronized, consistent_copying, or consistent_disconnected), you can issue the **access** parameter with the **stoprcconsistgrp** command to enable write access to the secondary volumes within that group. For a consistency group in the consistent_synchronized state, this command causes a consistency freeze.

The consistent_copying state is a consistent state. A consistency group in this state changes to the consistent_stopped state if it receives a **stoprcconsistgrp** command. Because the secondary change volume holds the consistent image, a stopped consistent_copying relationship might not have its secondary change volume deconfigured. This can be achieved by enabling access or completing synchronization so the secondary disk contains a consistent image. A relationship in consistent_copying or consistent_stopped accepts **stoprcrelationship -access** transition to idling state.

The consistent image that is present on the change volume is made accessible at the secondary volume and after the command completes the secondary volume can serve host read and write I/O operations.

If you specify stoprcconsistgrp -access for a consistency group in a consistent_copying state the last consistent image on all the relationships in that group is restored. This process starts a FlashCopy mapping with the secondary change volume for the secondary volume in each relationship, which might cause the command to fail.

The relationship's data is from a different point in time than the consistency group's data if:

1. The consistency group is in a consistent_copying state
2. You add a relationship to the group after the state became consistent_copying

Therefore, the relationship and consistency group are not mutually consistent, and attempting to stop and enable access to the consistency group results in an error. To fix this, let a background copy complete so the consistency group becomes consistent_synchronized) or remove the inconsistent relationship from the consistency group before you enable access. If you stop the consistency group without the **-access** parameter, the consistency group becomes consistent_stopped but the secondary change volumes continue to retain a consistent image.

A FlashCopy background copy operation begins to migrate the data for the consistent image from the change volume to the secondary volume. While the background copy operation is in progress, the change volume for the secondary volume remains in use.

It might be necessary to process I/O before the reverse FlashCopy map can be triggered, causing the enable access command to time out. In this case, the relationship delays changing to `idling` until the reverse map starts and write access is available. Read access to the consistent data remains available.

To stop `active-active` consistency groups:

- You specify **-access**
- The relationship's state is `consistent_copying`
- The relationship's status is `primary_offline`

Specify `stoprcconsistgrp -access` to obtain host read or write access to a volume in an `active-active` consistency group that contains an older but consistent image that might be needed in a disaster recovery scenario (the relationship has a `consistent_copying` state).

Any remote copy secondary volumes that are mapped to hosts of type `hide_secondary` are presented to the host if you specify **-access**. Paths to those volumes are revealed to the host, and a logical unit number (LUN) inventory changed unit attention is raised to report their availability.

Table 93 shows consistency group initial and final states:

Table 93. **stoprcconsistgrp** *consistency group states*

Initial state	Final state	Notes®
inconsistent_stopped	inconsistent_stopped	If **access** is specified, the command is rejected.
inconsistent_copying	inconsistent_stopped	If **access** is specified, the command is rejected with no effect and the relationship remains in the `inconsistent_copying` state.
consistent_stopped	consistent_stopped	If **access** is specified, the final state is `idling`.
consistent_synchronized	consistent_stopped	If **access** is specified, the final state is `idling`. If **access** is not specified, the final state is `consistent_stopped`.
consistent_copying	consistent_stopped	If **access** is specified, the final state is `idling`. If **access** is not specified, the final state is `consistent_stopped`.
idling	idling	Remains in `idling` state whether **access** is specified or not.
idling_disconnected	unchanged	If specified without **access**, the relationship or group remains in `idling_disconnected` state. If the clustered systems reconnect, the relationship/group is in either `inconsistent_stopped` or `consistent_stopped state`.
inconsistent_disconnected	inconsistent_stopped	The command is rejected, with or without the **access** flag.

Table 93. **stoprcconsistgrp** *consistency group states (continued)*

Initial state	Final state	Notes®
consistent_disconnected	consistent_stopped	The command is rejected if specified without **access**. If specified with **access**, the relationship or group moves to `idling_disconnected`.

An invocation example

```
stoprcconsistgrp rccopy1
```

The resulting output:

```
No feedback
```

stoprcrelationship

Use the **stoprcrelationship** command to stop the copy process for a Metro Mirror or Global Mirror stand-alone relationship. You can also use this command to enable write access to a consistent secondary volume that includes for an `active-active` relationship.

Syntax

```
►►── stoprcrelationship ── ─┬─────────┬─ ─┬─ rc_rel_id ───┬──────────────────────────────►◄
                            └─ -access ─┘   └─ rc_rel_name ─┘
```

Parameters

-access
(Optional) Specifies that the system allow write access to a consistent secondary volume.

rc_rel_id | *rc_rel_name*
(Required) Specifies the ID or the name of the relationship to stop all processing for.

Description

The **stoprcrelationship** command applies to a stand-alone relationship. The command is rejected if it is addressed to a relationship that is part of a consistency group. You can issue this command to stop a relationship that is copying from primary to secondary volumes.

Note: You cannot stop a relationship by using the **-access** parameter for a relationship if the primary and secondary volumes are different sizes.

If the relationship is in an inconsistent state, any copy operation stops and does not resume until you issue a **startrcrelationship** command. For a relationship in the `consistent_synchronized` state, this command causes a consistency freeze.

When a relationship is in a consistent state – in the `consistent_stopped`, `consistent_synchronized`, `consistent_copying`, or `consistent_disconnected` state – you can use the **access** parameter to enable write access to the secondary volume. Table 94 on page 581 provides consistency group initial and final states.

The `consistent_copying` state is a consistent state. A relationship in `consistent_copying` state transitions to `consistent_stopped` state when you specify **stoprcrelationship**. Because the secondary change volume holds the consistent image, a stopped `consistent_copying` relationship might not have its secondary change volume deconfigured. This can be achieved by enabling access or completing synchronization so the secondary disk contains a consistent image. A relationship in consistent_copying or consistent_stopped accepts **stoprcrelationship -access** transition to idling state.

The consistent image that is present on the change volume is made accessible at the secondary volume. After the command completes, the secondary volume can serve host read and write I/O.

A FlashCopy background copy operation migrates the data for the consistent image from the change volume to the secondary volume. While the background copy operation is in progress, the change volume for the secondary volume remains in use.

The enable access command can time out if there is I/O to process before the reverse FlashCopy map can be triggered. In this case, the relationship delays the transition to idling until the reverse map starts and write access is available. Read access to the consistent data remains available.

To stop active-active relationships:

- **-access** is specified
- The relationship's state is consistent_copying
- The relationship's status is primary_offline

Specify stoprelationship -access to obtain host read or write access to a volume in an active-active relationship that contains an older but consistent image that might be needed in a disaster recovery scenario (the relationship has a consistent_copying state).

Any remote copy secondary volumes that are mapped to hosts of type hide_secondary are presented to the host if you specify **-access**. Paths to those volumes appear to the host, and a logical unit number (LUN) inventory changed unit attention is raised to report their availability.

When you enable read or write access for a consistent_copying relationship, specify stoprcrelationship -access to restore a consistent image on the secondary change volume that is using a FlashCopy mapping. (Depending on how long this operation lasts the CLI command might delay.) This process fails if either the secondary volume or secondary change volume are offline. If you stop the relationship without specifying the **-access** parameter, the relationship becomes consistent_stopped and the secondary change volume is unchanged.

To enable access to a secondary volume that is not consistent, specify rmrcrelationship -force.

Table 94. **stoprcrelationship** *consistency group states*

Initial state	Final state	Notes
inconsistent_stopped	inconsistent_stopped	If **access** is specified, the command is rejected.
inconsistent_copying	inconsistent_stopped	If **access** is specified, the command is rejected with no effect and the relationship remains in the inconsistent_copying state.
consistent_stopped	consistent_stopped	If **access** is specified, the final state is idling.
consistent_synchronized	consistent_stopped	If **access** is specified, the final state is idling. If **access** is not specified, the final state is consistent_stopped.
consistent_copying	consistent_stopped	If **access** is specified, the final state is idling. If **access** is not specified, the final state is consistent_stopped.

*Table 94. **stoprcrelationship** consistency group states (continued)*

Initial state	Final state	Notes
idling	idling	Remains in `idling` state whether **access** is specified or not.
idling_disconnected	unchanged	If specified without **access**, the relationship or group remains in `idling_disconnected` state. If the clustered systems reconnect, the relationship or group is in either `inconsistent_stopped` or `consistent_stopped` state.
inconsistent_disconnected	inconsistent_stopped	The command is rejected, with or without the **access** flag.
consistent_disconnected	consistent_stopped	The command is rejected if specified without **access**. If specified with **access**, the relationship or group moves to `idling_disconnected` state.

An invocation example

```
stoprcrelationship rccopy1
```

The resulting output:

```
No feedback
```

switchrcconsistgrp

Use the **switchrcconsistgrp** command to reverse the roles of the primary and secondary volumes in a Metro Mirror or Global Mirror consistency group when that consistency group is in a consistent state. All the relationships in the consistency group are affected by this change.

Syntax

```
►►— switchrcconsistgrp — — -primary ─┬─ master ─┬─┬─ rc_consist_group_id ───┬──────────────────────►◄
                                     └─ aux ────┘ └─ rc_consist_group_name ─┘
```

Parameters

-primary master | aux
(Required) Specifies whether the master or auxiliary side of the relationships in the group become the primary volumes.

rc_consist_group_id | *rc_consist_group_name*
(Required) Specifies the ID or name of the consistency group to switch.

Description

This command applies to a consistency group. It is normally issued to reverse the roles of the primary and secondary volumes in a consistency group, perhaps as part of a failover process that is associated with a disaster recovery event.

Note: You cannot switch a consistency group if the primary and secondary volumes are different sizes.

Write access to the former primary volumes is lost and write access to the new primary volumes is acquired.

This command is successful when the consistency group is in a connected, consistent state, and when you reverse the direction of the relationships would not lead to a loss of consistency, for example, when the consistency group is consistent and synchronized. The consistency group must be in one of the following states in order for the **switchrcconsistgrp** command to process correctly:

- ConsistentSynchronized
- ConsistentStopped and Synchronized
- Idling and Synchronized

  Note: This command is rejected under any of the following conditions:
  - You switch consistency group relationship so that the new secondary becomes the target volume of an active FlashCopy mapping.
  - Any of the indicated secondary volumes (in the consistency group) are the target of an existing FlashCopy mapping.
  - Using Global Mirroring with the `multi` cycling mode

The consistency group moves to the ConsistentSynchronized state after the successful completion of this command. If you specify the **-primary** parameter and it is the same as the current primary, the command has no effect.

When the direction of the consistency group is changed, a volume that is a secondary volume in a remote copy relationship becomes a primary volume. In addition, a primary volume that is in a remote copy relationship becomes a secondary volume. If the resultant secondary volume is mapped to a host of type `hide_secondary`, it is no longer presented to that host. However, the mapping still exists for configuration purposes. If the volume that was a secondary volume before the switch is mapped to a host of type `hide_secondary`, it is presented to that host because it is no longer a secondary volume.

You cannot switch directions for an `active-active` consistency group.

An invocation example

```
switchrcconsistgrp -primary aux rccopy2
```

The resulting output:

```
No feedback
```

switchrcrelationship

Use the **switchrcrelationship** command to reverse the roles of primary and secondary volumes in a stand-alone Metro Mirror or Global Mirror relationship when that relationship is in a consistent state.

Syntax

```
►►── switchrcrelationship ── ── -primary ──┬─ master ─┬──┬─ rc_rel_id ───┬──────────────────►◄
                                           └─ aux ────┘  └─ rc_rel_name ─┘
```

Parameters

-primary master | aux
(Required) Specifies whether the master disk or the auxiliary disk is to be the primary.

rc_rel_id | rc_rel_name
(Required) Specifies the ID or the name of the relationship to switch.

Description

The **switchrcrelationship** command applies to a stand-alone relationship. It is rejected if it is used to try to switch a relationship that is part of a consistency group. It is normally issued to reverse the roles of the primary and secondary volume in a relationship perhaps as part of a failover process during a disaster recovery event.

Note: You cannot switch a relationship if the primary and secondary volumes are different sizes.

Write access to the old primary disk is lost. Write access to the new primary disk is acquired.

This command is successful when the relationship is in a connected, consistent state and when the direction of the relationship is reversed and it does not lead to a loss of consistency. This means that the relationship is consistent and synchronized. The relationship must be in one of the following states in order for the **switchrcrelationship** command to process correctly:

- ConsistentSynchronized
- ConsistentStopped and Synchronized
- Idling and Synchronized

  Note: A command in idling state is rejected if any of the indicated secondary volumes is the target of an existing FlashCopy map.

The relationship moves to the **ConsistentSynchronized** state after the successful completion of this command. If you specify the **-primary** parameter with the current primary, the command has no effect.

When the direction of the relationship is changed, a volume that is a secondary volume in a remote copy relationship becomes a primary volume, and a primary volume in a remote copy relationship becomes a secondary volume. If the resultant secondary volume is mapped to a host (type `hide_secondary`), it is no longer presented to that host. However, the mapping still exists for configuration purposes. If the volume that was a secondary volume before the switch is mapped to a host of type `hide_secondary`, it is presented to that host because it is no longer a secondary volume.

The **switchrcrelationship** command is rejected if you use Global Mirroring with the **multi** cycling mode.

You cannot switch directions for an `active-active` relationship.

An invocation example

```
switchrcrelationship -primary master rccopy2
```

The resulting output:

```
No feedback
```

Chapter 22. Migration commands

Use the migration commands to work with migration options for the system .

lsmigrate

Use the **lsmigrate** command to display the progress of all current data migration operations.

Syntax

```
►►── lsmigrate ── ┬─────────┬─ ┬──────────────────────┬──────────────────────────────►◄
                  └─ -nohdr ┘   └─ -delim ── delimiter ┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (**:**) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

If you use multiple threads to migrate data, the progress increases when all threads complete the migration of an extent. For large extent sizes with many threads, which can result in large increments in the percentage progress.

Description

This command displays information of all the migrations that are currently in progress.

Note: Only user-initiated migrations are reported by using this command. Easy Tier migrations are not included in the output.

An invocation example

```
lsmigrate -delim :
```

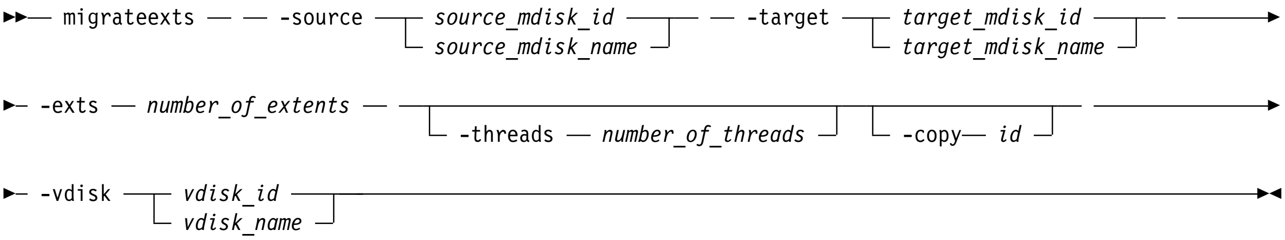
The resulting output:

```
migrate_type:MDisk_Group_Migration
progress:96
migrate_source_vdisk_index:33
migrate_target_mdisk_grp:4
max_thread_count:4
migrate_source_vdisk_copy_id:1
```

migrateexts

Use the **migrateexts** command to migrate extents from one managed disk to another.

Syntax



```
►►— migrateexts — — -source ─┬─ source_mdisk_id ───┬─ — -target ─┬─ target_mdisk_id ───┬─ ───────►
                             └─ source_mdisk_name ─┘              └─ target_mdisk_name ─┘

►— -exts — number_of_extents — ─┬─────────────────────────────────┬─┬──────────────┬─ ───────────►
                                └─ -threads — number_of_threads ─┘ └─ -copy— id ─┘

►— -vdisk ─┬─ vdisk_id ───┬──────────────────────────────────────────────────────────────────────►◄
           └─ vdisk_name ─┘
```

Parameters

-source *source_mdisk_id* | *source_mdisk_name*
(Required) Specifies the MDisk on which the extents currently reside.

-target *target_mdisk_id* | *target_mdisk_name*
(Required) Specifies the MDisk to migrate the extents to.

-exts *number_of_extents*
(Required) Specifies the number of extents to migrate.

-threads *number_of_threads*
(Optional) Specifies the number of threads to use while migrating these extents. You can specify 1 - 4 threads. The default number of threads is 4.

-copy *id*
(Required if the specified volume has more than one copy) Specifies the volume copy that the extents belong to.

-vdisk *vdisk_id* | *vdisk_name*
(Required) Specifies the volume that the extents belong to.

Description

This command migrates a given number of extents from the source volume and the managed disk that contains extents that are used to make up the volume. The target is a managed disk within the same storage pool.

You cannot specify this command for thin or compressed volume copies that are in data reduction storage pools.

If a large number of extents are being migrated, you can specify 1 - 4 threads. You can issue the **lsmigrate** command to check the progress of the migration.

The **migrateexts** command fails if there are insufficient free extents on the target managed disk. To avoid this problem, do not issue new commands that use extents until the extents migration is completed.

The **migrateexts** command fails if the target or source volume is offline, or if Easy Tier is active for the volume copy. Correct the offline condition before attempting to migrate the volume.

Note: Migration activity on a single managed disk is limited to a maximum of 4 concurrent operations. This limit does not take into account whether the managed disk is the source or the destination target. If more than four migrations are scheduled for a particular managed disk, further migration operations are queued pending the completion of one of the currently running migrations. If a migration operation is stopped for any reason, a queued migration task can be started. However, if a migration is suspended, the current migration continues to use resources and a pending migration is not started. For example, the following setup is a possible initial configuration:

- `MDiskGrp` 1 has volume 1 created in it
- `MDiskGrp` 2 has volume 2 created in it
- `MDiskGrp` 3 has only one MDisk

With the previous configuration, the following migration operations are started:

- Migration 1 migrates volume 1 from `MDiskGrp` 1 to `MDiskGrp` 3, running with 4 threads.
- Migration 2 migrates volume 2 from `MDiskGrp` 2 to `MDiskGrp` 3, running with 4 threads.

Due to the previous limitations, the two migration operations do not always run at the same speed. `MDiskGrp` 3 has only one MDisk and the two migration operations have a total of 8 threads that are trying to access the one MDisk. Four threads are active. The remaining threads are in standby mode waiting to access the MDisk.

Remember: This command cannot be used if the source MDisk is an SAS MDisk (which works in image mode only).

An invocation example

```
migrateexts -vdisk vdisk4 -source mdisk4 -exts
 64 -target mdisk6 -threads 4
```

The resulting output:

```
No feedback
```

migratetoimage

Use the **migratetoimage** command to migrate data from a volume (image mode or managed mode) onto a new image mode volume copy. The target disk does not have to be in the same storage pool as the source disk.

Syntax

```
►►─ migratetoimage ─┬──────────────┬─ ─ -vdisk ─┬─ source_vdisk_id ───┬─ ──────────────────►
                    └─ -copy─ id ──┘            └─ source_vdisk_name ─┘

►─┬──────────────────────────────┬─ ─ -mdisk ─┬─ unmanaged_target_mdisk_id ───┬──────────────►
  └─ -threads ─ number_of_threads ┘            └─ unmanaged_target_mdisk_name ─┘

►─┬───────────────────────────────┬─ ─ -mdiskgrp ─┬─ managed_disk_group_id ───┬────────────►◄
  └─ -tier ──┬─ tier0_flash ──────┤                └─ managed_disk_group_name ─┘
             ├─ tier1_flash ──────┤
             ├─ tier_enterprise ──┤
             └─ tier_nearline ────┘
```

Parameters

-vdisk *source_vdisk_id* | *name*
  (Required) Specifies the name or ID of the source volume to be migrated.

-copy *id*
  (Required if the specified volume has more than one copy) Specifies the volume copy to migrate from.

-threads *number_of_threads*
(Optional) Specifies the number of threads to use during the migration of extents. You can specify 1 - 4 threads. The default number of threads is 4.

-mdisk *unmanaged_target_mdisk_id* | *name*
(Required) Specifies the name of the MDisk to which the data must be migrated. This disk must be unmanaged and large enough to contain the data of the disk that is being migrated.

-mdiskgrp *managed_disk_group_id* | *name*
(Required) Specifies the storage pool into which the MDisk must be placed, after the migration has completed.

-tier *tier0_flash* | *tier1_flash* | *tier_enterprise* | *tier_nearline*
(Optional) Specifies the tier of the MDisk being added.

tier0_flash
Specifies a `tier0_flash` hard disk drive or an external MDisk for the newly discovered or external volume.

tier1_flash
Specifies an `tier1_flash` (or flash drive) hard disk drive or an external MDisk for the newly discovered or external volume.

tier_enterprise
Specifies a `tier_enterprise` hard disk drive or an external MDisk for the newly discovered or external volume.

tier_nearline
Specifies a `tier_nearline` hard disk drive or an external MDisk for the newly discovered or external volume.

Description

This command cannot be used to if the source volume copy is in a child pool or if the MDisk group that is specified is a child pool. This command does not work if the volume is fast formatting.

Note: You cannot migrate a volume or volume image between storage pools if cloud snapshot is enabled on the volume.

The **migratetoimage** command migrates the data of a user-specified volume by consolidating its extents (which might reside on one or more MDisks) onto the extents of the target MDisk that you specify. After migration is complete, the volume is classified as an image type volume, and the corresponding mdisk is classified as an image mode MDisk.

The managed disk that is specified as the target must be in an unmanaged state at the time that the command is run. Running this command results in the inclusion of the MDisk into the user-specified storage pool.

You cannot specify **migratetoimage** if the target or source volume is offline. Correct the offline condition before you migrate the volume.

Remember: This command cannot be used on a volume that is owned by a filesystem or if the source MDisk is an SAS MDisk (which works in image mode only).

If the volume (or volume copy) is a target of a FlashCopy mapping with a source volume in an `active-active` relationship the new storage pool must be in the same site as the source volume. If the volume is in an `active-active` relationship the new storage pool must be located in the same site as the source volume. Additionally, the site information for the MDisk being added must be well-defined and match the site information for other MDisks in the storage pool.

Note: You cannot migrate date from a volume if the target volume's formatting attribute value is `yes`.

You cannot specify **`migratetoimage`** to migrate a thin or compressed volume in a data reduction storage pool to an image mode volume. You must replicate the volume to a fully allocated, thin, or compressed volume copy in a standard storage pool before you migrate a thin or compressed volume that is in a data reduction pool from one clustered system to another clustered system.

An encryption key cannot be used when migrating an image mode MDisk. To use encryption (when the MDisk has an encryption key), the MDisk must be self-encrypting.

An invocation example

The following example specifies that the user wants to migrate the data from `vdisk0` onto `mdisk5` and that the MDisk must be put into the storage pool `mdgrp2`.

```
migratetoimage -vdisk vdisk0 -mdisk mdisk5 -mdiskgrp mdgrp2 -tier tier_nearline
```

The resulting output:

```
No feedback
```

migratevdisk

Use the **`migratevdisk`** command to migrate an entire volume from one storage pool to another storage pool.

Syntax

```
►►— migratevdisk — — -mdiskgrp ─┬─ mdisk_group_id ───┬─ ─┬─────────────────────────────────────┬──────►
                                └─ mdisk_group_name ─┘   └─ -threads — number_of_threads ─┘

►─ ─┬───────────────┬─ — -vdisk ─┬─ vdisk_id ───┬────────────────────────────────────────────────────►◄
    └─ -copy— id ─┘              └─ vdisk_name ─┘
```

Parameters

-mdiskgrp *mdisk_group_id* | *mdisk_group_name*
(Required) Specifies the new storage pool ID or name.

-threads *number_of_threads*
(Optional) Specifies the number of threads to use during the migration of these extents. You can specify 1 - 4 threads. The default number of threads is 4.

-copy *id*
(Required if the specified volume has more than one copy) Specifies the volume copy to migrate.

-vdisk *vdisk_id* | *vdisk_name*
(Required) Specifies the volume ID or name to migrate in to a new storage pool.

Description

The **`migratevdisk`** command migrates the specified volume into a new storage pool; all the extents that make up the volume are migrated onto free extents in the new storage pool.

You can reassign a volume from a:

- Child pool to its parent pool
- Parent pool to one of its child pools
- Between the child pools in the same parent pool

- Between two parent pools

Note: You cannot migrate a volume between storage pools if cloud snapshot is enabled on the volume.

You can issue the **lsmigrate** command to view the progress of the migration.

The process can be prioritized by specifying the number of threads to use during the migration. Using only one thread puts the least background load on the system.

The **migratevdisk** command fails if there are insufficient free extents on the targeted storage pool for the duration of the command. To avoid this problem, do not issue new commands that use extents until the volume migration is completed.

The **migratevdisk** command fails if the target volume or source volume is offline. Correct the offline condition before you attempt to migrate the volume.

Remember: You cannot specify this command:
- For volumes that are owned by a file system.
- If the source MDisk is an SAS MDisk (which works in image mode only).
- If the volume that is being migrated is thin-provisioned or compressed and in a data reduction pool.
- If the target pool is a data reduction pool and the volume that is being migrated is either thin-provisioned or compressed.

For these volume types, you must create a volume copy in the destination pool by using volume mirroring to perform the migration. For more information, see the **addvdiskcopy** or **addvolumecopy** command.

If the volume (or volume copy) is a target of a FlashCopy mapping with a source volume in an `active-active`, relationship the new storage pool must be in the same site as the source volume. If the volume is in an `active-active` relationship, the new storage pool must be located in the same site as the source volume.

When the volume is being migrated from a parent pool to another parent pool, the information is moved (unchanged), whether or not one or other is encrypted. The parent pool and the child pool cannot have an encryption key (or else the child pool would have failed during creation).
- A parent pool to parent pool migration is allowed in all cases.
- A parent pool to child pool migration is not allowed if child has encryption key.
- A child pool to parent pool or child pool is not allowed if either child pool has an encryption key.

An invocation example

```
migratevdisk -vdisk 4 -mdiskgrp Group0 -threads 2
```

The resulting output:

```
No feedback
```

Chapter 23. Service mode commands (Discontinued)

The service mode commands are discontinued.

applysoftware (Discontinued)

Attention: The **svcservicemodetask applysoftware** command is discontinued. Use the **satask installsoftware** command instead.

Discontinued.

svcservicemodetask cleardumps (Discontinued)

Attention: The **svcservicemodetask cleardumps** command is discontinued. Use the **cleardumps** command instead.

Discontinued.

svcservicemodetask dumperrlog (Discontinued)

Attention: The **svcservicemodetask dumperrlog** command is discontinued. Use the **dumperrlog** command instead.

Discontinued.

exit (Discontinued)

Attention: The **svcservicemodetask exit** command is discontinued. Use the **satask stopservice** command instead.

Chapter 24. Service mode information commands (Discontinued)

The service mode information commands are discontinued.

ls2145dumps (Discontinued)

The **svcservicemodeinfo ls2145dumps** command is discontinued. Use the **lsdumps** command to display a list of files in a particular dumps directory.

lscimomdumps (Discontinued)

The **svcservicemodeinfo lscimomdumps** command is discontinued. Use the **lsdumps** command to display a list of files in a particular dumps directory.

lsclustervpd (Discontinued)

Attention: The **svcservicemodeinfo lsclustervpd** command is discontinued. Use the **sainfo lsservicestatus** command instead.

lserrlogdumps (Discontinued)

The **svcservicemodeinfo lserrlogdumps** command is discontinued. Use the **lsdumps** command to display a list of files in a particular dumps directory.

lsfeaturedumps (Discontinued)

The **svcservicemodeinfo lsfeaturedumps** command is discontinued. Use the **lsdumps** command to display a list of files in a particular dumps directory.

lsiostatsdumps (Discontinued)

The **svcservicemodeinfo lsiostatsdumps** command is discontinued. Use the **lsdumps** command to display a list of files in a particular dumps directory.

lsiotracedumps (Discontinued)

The **svcservicemodeinfo lsiotracedumps** command is discontinued. Use the **lsdumps** command to display a list of files in a particular dumps directory.

lsmdiskdumps (Discontinued)

The **svcservicemodeinfo lsmdiskdumps** command is discontinued. Use the **lsdumps** command to display a list of files in a particular dumps directory.

lssoftwaredumps (Discontinued)

The **svcservicemodeinfo lssoftwaredumps** command is discontinued. Use the **lsdumps** command to display a list of files in a particular dumps directory.

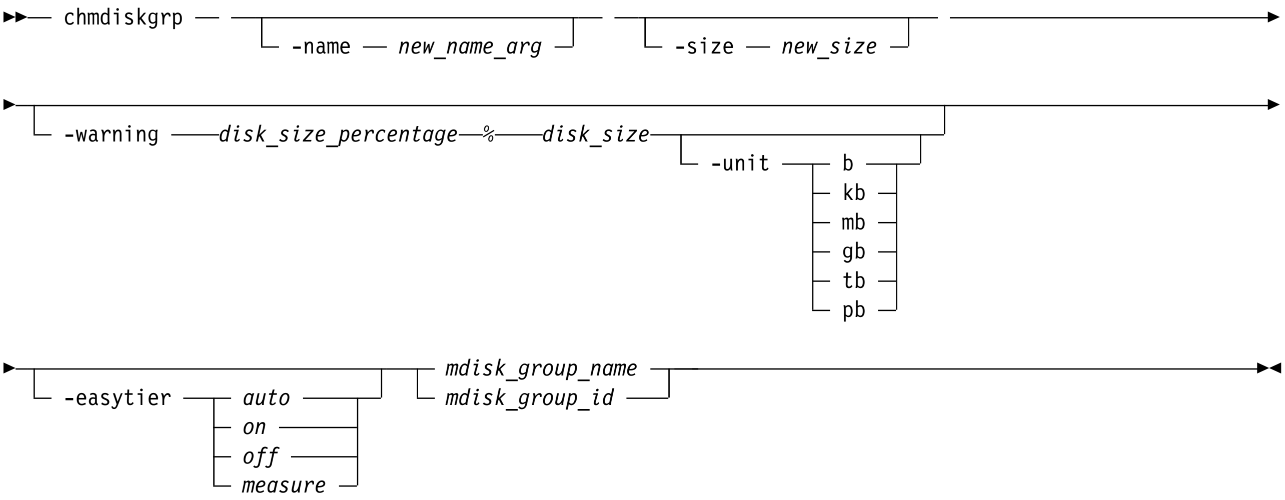
Chapter 25. Storage pool commands

Use the storage pool commands to work with storage pool options on the system.

chmdiskgrp

Use the **chmdiskgrp** command to modify the name that is assigned to a storage pool or to set the warning threshold for the storage pool.

Syntax



Parameters

-name *new_name_arg*
(Optional) Specifies the new name of the storage pool.

-warning *disk_size* | *disk_size_percentage%*
(Optional) Sets a threshold at which a warning is generated. The warning is generated the first time that the threshold is exceeded by the used-disk capacity in the storage pool. You can specify a *disk_size* integer, which defaults to megabytes (MB) unless the **-unit** parameter is specified; or you can specify a *disk_size*%, which is a percentage of the storage pool size. To disable warnings, specify **0** or **0**%.

-size *new_size*
(Optional) Specifies the new size of a child pool.

Note: This parameter cannot be used with parent pools. Use **addmdisk** or **rmmdisk** to change storage pool capacity.

-unit b | kb | mb | gb | tb | pb
(Optional) Specifies the data units for the **-warning** parameter.

-easytier *auto* | *on* | *off* | *measure*
(Optional) Specifies if the Easy Tier function is on or off for this storage pool, or if it is automatically determined. **-easytier** is `active` in storage pools with multiple tiers and is `balance` with single tiers.

Note: **-easytier** must be followed by one of the following:

- If **-easytier** is set to *auto*, SAN Volume Controller automatically enables Easy Tier functions when the storage pool contains MDisks from more than one tier, and enables automatic rebalancing when the storage pool contains MDisks from only one tier.
- If **-easytier** is set to *on*, then Easy Tier functions are active.
- If **-easytier** is set to *off*, then Easy Tier functions are inactive.
- If **-easytier** is set to *measure* Easy Tier statistics are collected but Easy Tier management is disabled. (No extents are moved by Easy Tier.)

auto equates to:

- *on* if Easy Tier is licensed or no license is required
- *off* if Easy Tier is not licensed and a license is required

Specifying `-easytier` *on* enables Easy Tier:

- Management of both single-tier and multi-tier pools
- Auto rebalance

Extents are moved to balance the I/O load on the MDisks in the pool.

mdisk_group_id | *mdisk_group_name*
(Required) Specifies the ID or name of the storage pool to modify.

Description

Table 95. Parameter differences for child pools and parent pools

Parameter	Child pool usage	Storage pool usage
-name	Optional	Optional
-easytier	Cannot be used with child pools	Optional
-size	Optional	Cannot be used with parent pools
-unit	Optional	Optional
-warning	Optional	Optional

This command modifies the name, or label, assigned to a given storage pool. You can use the new name to refer to the storage pool.

The command can also be used to set the warning threshold for the storage pool. The warning threshold is the threshold which a warning is generated when it is exceeded by the used-disk capacity in the storage pool.

You can also use this command to change other settings for parent pools and child pools.

An invocation example

```
chmdiskgrp -name testmdiskgrp -easytier on Group0
```

The resulting output:

```
No feedback
```

An invocation example

```
chmdiskgrp -size 100 -unit tb mypool
```

The resulting output:

```
No feedback
```

lsfreeextents

Use the **lsfreeextents** command to list the number of free extents that are available on a specified MDisk.

Syntax

```
►►── lsfreeextents ── ┬──────────┬─ ┬──────────────────────┬─ ┬─ mdisk_id ───┬───────────────►◄
                      └─ -nohdr ─┘  └─ -delim ── delimiter ─┘  └─ mdisk_name ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim **:** on the command line, the colon character (**:**) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

mdisk_id | *mdisk_name*
(Required) Specifies the ID or the name of the MDisk for which you want to know the number of free extents.

Description

This command displays a count of the number of free extents on the specified MDisk.

An invocation example

```
lsfreeextents 2
```

The resulting output:

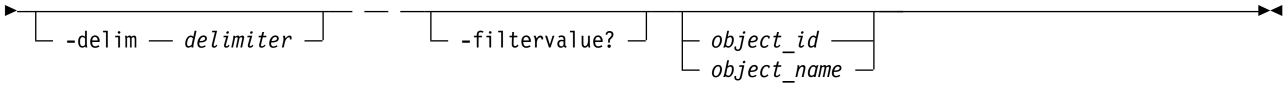
```
id 2
number_of_extents 4372
```

lsmdiskgrp

Use the **lsmdiskgrp** command to display a concise list or a detailed view of storage pools that are visible to the clustered system (system).

Syntax

```
►►── lsmdiskgrp ── ┬───────────────────────────────────┬─ ┬──────────┬─ ┬──────────┬─ ─────────►
                   └─ -filtervalue ── attribute=value ─┘  └─ -nohdr ─┘  └─ -bytes ─┘
```



Parameters

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are returned. If a capacity is specified, the units must also be included.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards when you use the command-line interface (CLI):

- The wildcard character is an asterisk (*).
- The command can contain a maximum of one wildcard, which must be the first or last character in the string.
- When you use a wildcard, you must enclose the filter entry within double quotation marks (""), as shown in the following command:

```
lsmdiskgrp -filtervalue  "name=md*"
```

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If no data is available to be displayed, headings are not displayed.

-bytes
(Optional) Specifies that you want the report to display all capacities as bytes.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

object_id | *object_name*
(Optional) Specifies the name or ID of an object. When you use this parameter, the detailed view of the specific object is returned and any value that is specified by the **-filtervalue** parameter is ignored. If you do not specify the *object_id* | *object_name* parameter, the concise view of all objects that match the filtering requirements that are specified by the **-filtervalue** parameter are displayed.

-filtervalue?
(Optional) Displays a list of valid filter attributes. The valid filters for the **lsmdiskgrp** command are the following values:

- name
- id
- mdisk_count
- vdisk_count
- status
- storage_pool_id
- easy_tier
- easy_tier_status

- site_id
- site_name
- parent_mdisk_grp_id
- parent_mdisk_grp_name
- child_mdisk_grp_count
- type
- encrypt
- owner_type
- data_reduction

Description

This command returns a concise list or a detailed view of storage pools visible to the system.

Command output includes values for the following attributes:

status The state of the MDisk with the highest-priority status in the group, excluding image mode MDisks.

VDisk_count
The number of volume copies that are in the storage pool.

capacity
The total amount of MDisk storage that is assigned to the storage pool.

extent_size
The sizes of the extents for this group are the following values: 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, or 8192 (MB).

free_capacity
2 The amount of MDisk storage that is immediately available. Additionally, `reclaimable_capacity`
2 can eventually become available.

real_capacity
2 The total MDisk storage capacity assigned to volume copies.

2 **Note:** It includes `reclaimable_capacity`.

virtual_capacity
2 The total host mappable capacity of all volume copies in the storage pool.

used_capacity
2 The amount of data that is stored on MDisks. Fully-allocated volumes contribute their entire
2 capacity.

2 **Note:** It includes `reclaimable_capacity`.

overallocation
Expressed as a percentage, the ratio of the `virtual_capacity` value to the capacity. A storage pool overallocation of over 100 is only possible if you configure thin-provisioned volume copies.

warning
This field is a percentage. A warning is generated when the amount of space in the storage pool that is assigned exceeds this level.

easy_tier
This value is set by the user and determines whether Easy Tier is permitted to manage the pool.

Note: The values are:

1. `on` indicates that Easy Tier actively manages the extents (including single-tier storage pools), and the Easy Tier status must be `active` - unless a license is required.
2. `off` indicates that Easy Tier does not actively manage the extents, and the Easy Tier status must be `inactive` .
3. `auto` indicates that the value of Easy Tier status is determined by the number of tiers in a storage pool.

   Note: The following values apply to `auto`:
   - `on` if Easy Tier is licensed or no license is required.
   - `off` if Easy Tier requires a license and none exists.
4. `measure` indicates that Easy Tier s collects statistics on that storage pool but does not move any extents in the storage pool.

easy_tier_status

This field indicates whether the Easy Tier functions are active on a storage pool.

- `active` indicates that a pool is being managed by Easy Tier to provide tier management performance-based pool balancing (for example, extents can be moved).
- `inactive` indicates that Easy Tier is inactive.
- `measured` indicates that Easy Tier statistics are being collected but no Easy Tier management is detected.
- `balanced` indicates that a pool is being managed by Easy Tier to provide performance-based pool balancing (for example, extents can be moved).

The following table describes the storage pool Easy Tier settings.

Table 96. Easy Tier settings for storage pools and volumes

Storage pool Easy Tier setting	Number of tiers in the storage pool	Volume copy Easy Tier setting	Volume copy Easy Tier status
Off	One	Off	inactive (see note 1 on page 601)
Off	One	On	inactive (see note 1 on page 601)
Off	Two	Off	inactive (see note 1 on page 601)
Off	Two	On	inactive (see note 1 on page 601)
Measure	One	Off	measured (see note 2 on page 601)
Measure	One	On	measured (see note 2 on page 601)
Measure	Two	Off	measured (see note 2 on page 601)
Measure	Two	On	measured (see note 2 on page 601)
Auto	One	Off	measured (see note 2 on page 601)
Auto	One	On	balanced (see note 3 on page 601)
Auto	Two	Off	measured (see note 2 on page 601)

Table 96. Easy Tier settings for storage pools and volumes (continued)

Storage pool Easy Tier setting	Number of tiers in the storage pool	Volume copy Easy Tier setting	Volume copy Easy Tier status
Auto	Two	On	active (see note 4)
On	One	Off	balanced (see note 3)
On	One	On	balanced (see note 3)
On	Two	Off	measured (see note 2)
On	Two	On	active (see note 4)

Notes:

1. When the volume copy status is `inactive`, no Easy Tier functions are enabled for that volume copy.
2. When the volume copy status is `measured`, the Easy Tier function collects usage statistics for the volume but automatic data placement is not active.
3. When the volume copy status is `balanced`, the Easy Tier function enables performance-based pool balancing for that volume copy.
4. When the volume copy status is `active`, the Easy Tier function operates in automatic data placement mode for that volume.

If the volume copy is in image or sequential mode or is being migrated, the volume copy Easy Tier status is `measured` instead of `active`.

The default Easy Tier setting for a storage pool is `auto`, and the default Easy Tier setting for a volume copy is `on`. Thus, Easy Tier functions except pool performance balancing are disabled for storage pools with a single tier, and that automatic data placement mode is enabled for all striped volume copies in a storage pool with two or more tiers.

tier Indicates which tier information is being reported. The values are:

- `tier0_flash`
- `tier1_flash`
- `tier_enterprise`
- `tier_nearline`

tier_mdisk_count
Indicates the number of MDisks in the tier.

tier_capacity
2 The amount of MDisk storage in this tier that is assigned to the storage pool.

tier_free_capacity
2 The amount of MDisk storage in this tier that has not been assigned.

compression_active
Indicates whether any compressed volume copies are in the storage pool. This field is blank for storage pools that are data reduction pools.

compression_virtual_capacity
2 Indicates the total virtual capacity for all compressed volume copies in regular storage pools. This
2 field reports 0.00MB for data reduction pools.

compression_compressed_capacity
2 Indicates the total used capacity for all compressed volume copies in regular storage pools. This
2 field reports 0.00MB for data reduction pools.

compression_uncompressed_capacity
Indicates the total uncompressed used capacity for all compressed volume copies in regular
2 storage pools. This field reports 0.00MB for data reduction pools.

site_id
Indicates the site value for the storage pool group. This numeric value is `1`, `2`, `3`, or blank.

site_name
Indicates the site name for the storage pool. This value is an alphanumeric value or is blank.

parent_mdisk_grp_id
Indicates the storage pool group ID. This value is a numeric string (in the range 0 - 127 characters) or blank.

parent_mdisk_grp_name
Indicates the storage pool group name. This value is an alphanumeric string (in the range 1 - 63 characters) or blank.

child_mdisk_grp_count
Indicates the number of child pools in the parent pools. This value is a numeric string (in the range 0 - 127 characters) or blank.

child_mdisk_grp_capacity
Indicates the total amount of space that is reserved for child pools.

type Indicates the MDisk group type. The values are `parent` and `child_thick`.

encrypt
Indicates whether the data that is stored on the MDisk group is encrypted or not encrypted. The values are:

- `yes` if the pool has an encryption key.
- `yes` if the pool does not have an encryption key (and the pool contains MDisks and all are encrypted).
- `no` if the pool does not have an encryption key (and the pool contains MDisks and at least one is not encrypted).
- Blank if the pool does not have an encryption key (and the pool has no MDisks).

The following define the status fields, from lowest to highest priority:

Online
Indicates that the storage pool is online and available.

Offline
Indicates that all paths to the storage pool are lost.

owner_type
Indicates the type of owning object, such as a file system or application. This attribute is an alphanumeric string up to 20 characters in length.

owner_id
Indicates an identifier for the owning object. It is represented by a number and is blank if no owning object exists.

owner_name
Indicates the name for the object that owns the volume. This attribute is an alphanumeric string up to 63 characters in length or is blank.

data_reduction
Indicates that the storage pool is a data reduction pool. The values are `yes` or `no`.

physical_capacity
Indicates the total physical capacity of MDisks that belong to this storage pool. For any disks that do not display their physical capacity, displays the logical capacity value. The value must be a number (indicated in units) that is rounded to two decimal places.

physical_free_capacity

Indicates the total free physical capacity of MDisks that belong to this storage pool. For any disks that do not display their physical capacity, displays the logical capacity value. The value must be a number (indicated in units) that is rounded to two decimal places.

shared_resources

Indicates that one or more MDisks in this storage pool shares a resource provisioning group with an MDisk in another storage pool. It indicates cross contamination over-provisioning of physical resources between the storage pools. The value must be `yes` or `no`.

reclaimable_capacity

2 The approximate amount of storage in a data reduction pool that the system can eventually make
2 available.

used_capacity_before_reduction

2 The data that is stored on non-fully-allocated volume copies in a data reduction pool. It indicates
2 the capacity before compression and deduplication.

used_capacity_after_reduction

2 The data that is stored on MDisks for non-fully-allocated volume copies in a data reduction pool.
2 It indicates the capacity after compression and deduplication.

deduplication_capacity_saving

1 The capacity that is saved by deduplication before compression in a data reduction pool.

overhead_capacity

2 The MDisk capacity that is reserved for internal usage.

compression_opportunity

2 The total capacity of all compressed volume copies in a data reduction pool.

2 **Note:** It excludes deduplication_capacity_saving.

deduplication_opportunity

2 The total **`used_capacity_before_reduction`** of all volume copies in a data reduction pool that are
2 data deduplication enabled.

A concise invocation example

```
lsmdiskgrp -delim :
```

The following concise output is displayed:

```
id:name:status:mdisk_count:vdisk_count:capacity:extent_size:free_capacity:virtual_capacity:used_capacity:
real_capacity:overallocation:warning:easy_tier:easy_tier_status:compression_active:
compression_virtual_capacity:compression_compressed_capacity:compression_uncompressed_capacity:
parent_mdisk_grp_id:parent_mdisk_grp_name:child_mdisk_grp_count:child_mdisk_grp_capacity:type:encrypt:
owner_type:site_id:site_name:data_reduction:used_capacity_before_reduction:used_capacity_after_reduction:
deduplication_capacity_saving:reclaimable_capacity

0:mdiskgrp0:online:2:0:399.00GB:256:399.00GB:0.00MB:0.00MB:
0.00MB:0:0:auto:balanced:no:
0.00MB:0.00MB:0.00MB:
0:mdiskgrp0:0:0.00MB:parent:no:
none:::no:0.00MB:0.00MB:
0.00MB:0.00MB

1:A9000:online:3:2:584.69GB:64:518.69GB:66.00GB:66.00GB:
66.00GB:11:0:auto:balanced:no:
0.00MB:0.00MB:0.00MB:
1:A9000:0:0.00MB:parent:no:
none:::no:0.00MB:0.00MB:
0.00MB:0.00MB

2:Storwize:online:3:2:592.50GB:64:526.50GB:66.00GB:66.00GB:
```

```
66.00GB:11:0:auto:balanced:no:
0.00MB:0.00MB:0.00MB:
2:Storwize:0:0.00MB:parent:no:
none:::no:0.00MB:0.00MB:
0.00MB:0.00MB
```

A detailed invocation example for a storage pool with one tier

```
lsmdiskgrp -delim : mdiskgrp1
```

The following output is displayed:

```
id:1
name:mdiskgrp1
status:online
mdisk_count:4
vdisk_count:6
capacity:200GB
extent_size:16
free_capacity:100GB
virtual_capacity:400.00GB
used_capacity:75.00GB
real_capacity:100.00GB
overallocation:200
warning:80
easy_tier:on
easy_tier_status:active
tier:ssd
tier_mdisk_count:0
tier_capacity: 0.00MB
tier_free_capacity:0.00MB
tier tier0_flash
tier_mdisk_count 1
tier_capacity 1.63TB
tier_free_capacity 1.63TB
tier tier1_flash
tier_mdisk_count 1
tier_capacity 1.63TB
tier_free_capacity 1.63TB
tier tier_enterprise
tier_mdisk_count 0
tier_capacity 0.00MB
tier_free_capacity 0.00MB
tier tier_nearline
tier_mdisk_count 0
tier_capacity 0.00MB
tier_free_capacity 0.00MB
compression_active:yes
compression_virtual_capacity:1000.00MB
compression_compressed_capacity:0.41MB
compression_uncompressed_capacity:512.05MB
site_id:3
site_name:Quorum
parent_mdisk_grp_id:3
parent_mdisk_grp_name:sisfyle
child_mdisk_grp_count:0
child_mdisk_grp_capacity:0.00MB
type:child_thick
encrypt:no
owner_type vvol_child_pool
owner_id
owner_name
physical_capacity:1.23TB
physical_free_capacity:1.11TB
shared_resources:yes
data_reduction:yes
reclaimable_capacity:0.00MB
```

```
used_capacity_before_reduction:24.03GB
used_capacity_after_reduction:18.17GB
deduplication_capacity_saving:3.22GB
overhead_capacity
```

A detailed invocation example for a storage pool with two tiers

```
lsmdiskgrp -delim : mdiskgrp2
```

The following output is displayed:

```
id:2
name:mdiskgrp2
status:online
mdisk_count:8
vdisk_count:6
capacity:200GB
extent_size:16
free_capacity:100GB
virtual_capacity:400.00GB
used_capacity:75.00GB
real_capacity:100.00GB
overallocation:200
warning:80
easy_tier:auto
easy_tier_status:active
tier:ssd
tier_mdisk_count:2
tier_capacity:20.00GB
tier_free_capacity:0.00MB
tier tier0_flash
tier_mdisk_count 1
tier_capacity 1.63TB
tier_free_capacity 1.63TB
tier tier1_flash
tier_mdisk_count 1
tier_capacity 1.63TB
tier_free_capacity 1.63TB
tier tier_enterprise
tier_mdisk_count 0
tier_capacity 0.00MB
tier_free_capacity 0.00MB
tier tier_nearline
tier_mdisk_count 0
tier_capacity 0.00MB
tier_free_capacity 0.00MB

tier_mdisk_count:6
tier_capacity:180.00GB
tier_free_capacity:100.00GB
tier:ri_ssd
tier_mdisk_count:
tier_capacity:
tier_free_capacity:
compression_active:yes
compression_virtual_capacity:1000.00MB
compression_compressed_capacity:0.41MB
compression_uncompressed_capacity:512.05MB
site_id:2
site_name:POK
parent_mdisk_grp_id:2
parent_mdisk_grp_name:sysfile
child_mdisk_grp_count:0
child_mdisk_grp_capacity:0.00MB
type:child_thick
owner_type vvol_child_pool
owner_id
owner_name
```

```
physical_capacity:1.63TB
physical_free_capacity:1.52TB
shared_resources:no
data_reduction:yes
reclaimable_capacity:15.00MB
used_capacity_before_reduction
used_capacity_after_reduction
overhead_capacity
```

A detailed invocation example for a storage pool with three tiers

```
lsmdiskgrp -delim : mdiskgrp1
```

The following output is displayed:

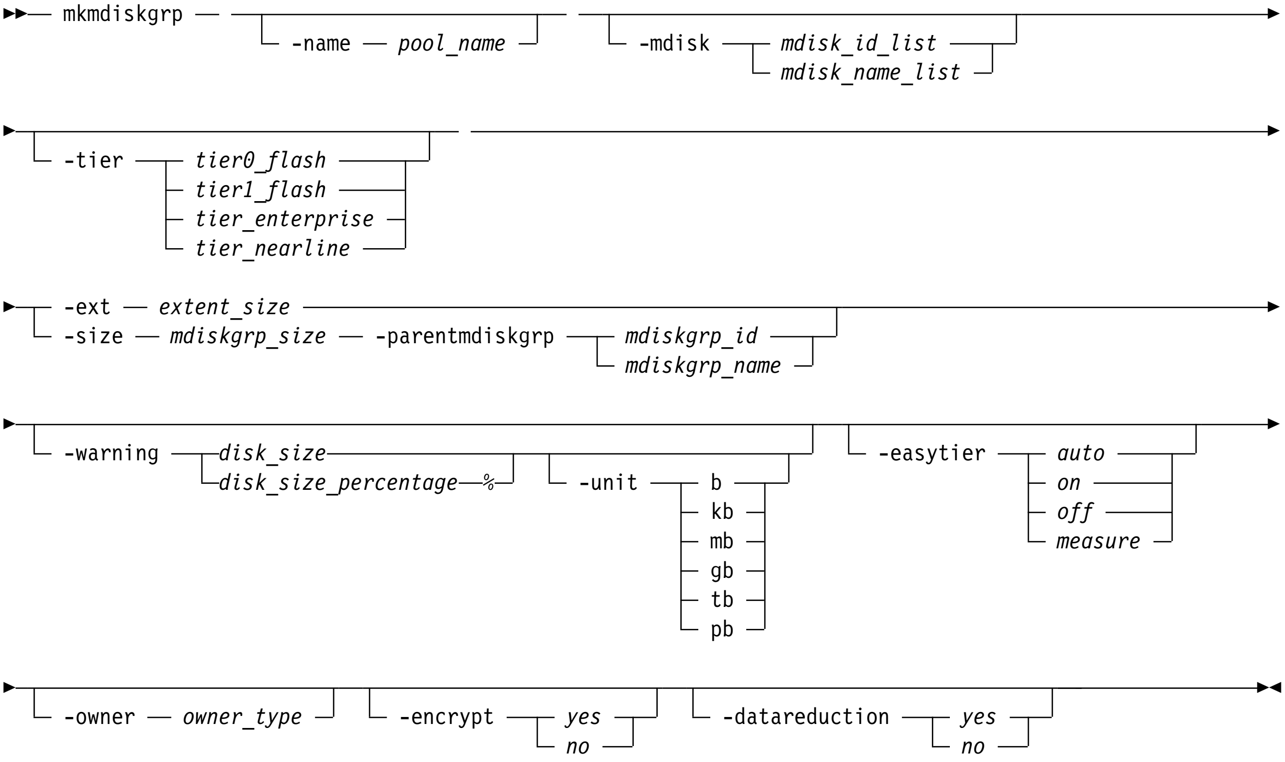
```
id:1
name:mdiskgrp1
status:online
mdisk_count:4
vdisk_count:6
capacity:200.00GB
extent_size:16
free_capacity:100.00GB
virtual_capacity:400.00GB
used_capacity:75.00GB
real_capacity:100.00GB
overallocation:200
warning:80
easy_tier:auto
easy_tier_status:inactive
tier tier0_flash
tier_mdisk_count 1
tier_capacity 1.63TB
tier_free_capacity 1.63TB
tier tier1_flash
tier_mdisk_count 1
tier_capacity 1.63TB
tier_free_capacity 1.63TB
tier tier_enterprise
tier_mdisk_count 0
tier_capacity 0.00MB
tier_free_capacity 0.00MB
tier tier_nearline
tier_mdisk_count 0
tier_capacity 0.00MB
tier_free_capacity 0.00MB

compression_active:no
compression_virtual_capacity:0.00MB
compression_compressed_capacity:0.00MB
compression_uncompressed_capacity:0.00MB
site_id:2
site_name:POK
parent_mdisk_grp_id:1
parent_mdisk_grp_name:filesys
child_mdisk_grp_count:0
child_mdisk_grp_capacity:0.00MB
type:child_thick

owner_type vvol_child_pool
owner_id
owner_name
physical_capacity:1.63TB
physical_free_capacity:1.52TB
shared_resources:no
data_reduction:yes
```

```
reclaimable_capacity:20.00MB
used_capacity_before_reduction
used_capacity_after_reduction
overhead_capacity
```

mkmdiskgrp

Use the **mkmdiskgrp** command to create a new storage pool.

Syntax



Parameters

-name *pool_name*
(Optional) Specifies a name to assign to the new pool.

-mdisk *mdisk_id_list* | *mdisk_name_list*
(Optional) Specifies a colon-separated list of managed disk IDs or names to add to the storage pool. You can create an empty storage pool by not specifying the **-mdisk** parameter.

-tier *tier0_flash* | *tier1_flash* | *tier_enterprise* | *tier_nearline*
(Optional) Specifies the tier of the MDisk or MDisks being added.

tier0_flash
Specifies a `tier0_flash` hard disk drive or an external MDisk for the newly discovered or external volume.

tier1_flash
Specifies an `tier1_flash` (or flash drive) hard disk drive or an external MDisk for the newly discovered or external volume.

tier_enterprise
Specifies a `tier_enterprise` hard disk drive or an external MDisk for the newly discovered or external volume.

tier_nearline
Specifies a `tier_nearline` hard disk drive or an external MDisk for the newly discovered or external volume.

If you do not specify a tier, the current tier value of the MDisk is retained. The default value for an external MDisk is *enterprise*.

-ext *extent_size*
(Required) Specifies the size of the extents for this group in MB. The **ext** parameter must have one of the following values: `16`, `32`, `64`, `128`, `256`, `512`, `1024`, `2048`, `4096`, or `8192` (MB).

-size *mdiskgrp_size*
(Optional) Specifies the child pool capacity. The value must be a numeric value (and an integer multiple of the extent size).

-parentmdiskgrp *mdiskgrp_id* | *mdiskgrp_name*
(Optional) Specifies the parent pool from which the volume extents of the child pool are allocated when you create a child pool. The value must be an *mdiskgrp_id* or *mdiskgrp_name*.

-warning *disk_size* | *disk_size_percentage%*
(Optional) Generates a warning when the used disk capacity in the storage pool first exceeds the specified threshold. You can specify a *disk_size* integer, which defaults to megabytes (MB) unless the **-unit** parameter is specified; or you can specify a *disk_size*%, which is a percentage of the storage pool size. To disable warnings, specify `0` or `0%`. The default value is `0`.

-unit b | kb | mb | gb | tb | pb
(Optional) Specifies the data units for the **-warning** parameter.

-easytier *on* | *off* | *auto* | *measure*
(Optional) Specifies if the Easy Tier function is active for this storage pool, or if it is automatically determined. *auto* is the default value. **-easytier** is `active` in storage pools with multiple tiers and is `balance` with single tiers.

Note:

- If **-easytier** is set to *auto*, the system automatically enables Easy Tier functions when the storage pool contains MDisk from more than one tier, and enables automatic rebalancing when the storage pool contains an MDisk from only one tier.
- If **-easytier** is set to *on*, then Easy Tier functions are active.
- If **-easytier** is set to *off*, then Easy Tier functions are inactive.
- If **-easytier** is set to *measure* Easy Tier statistics are collected but Easy Tier management is disabled. (No extents are moved by Easy Tier).

auto equates to:

- *on* if Easy Tier is licensed or no license is required.
- *off* if Easy Tier is not licensed and a license is required.

Specifying `-easytier` *on* enables Easy Tier:

- Management of both single-tier and multitier pools
- Auto rebalance

Extents are moved to balance the I/O load on the MDisks in the pool.

-owner *owner_type*
(Optional) Specifies the owner type. The value must be `vvol_child_pool`.

-encrypt *yes* | *no*
(Optional) Specifies the encryption status for this storage pool. The values are yes or no.

Remember:

- If you do not specify **-encrypt** and encryption is enabled, the system defaults to -encrypt yes (the default encryption setting).
- When you create a child pool in an encrypted parent pool, the value must not be no. (All other permutations are permitted).

-datareduction *yes* | *no*
(Optional) Specifies whether the storage pool is a data reduction storage pool. The values are yes or no. A value of no specifies that the storage pool is a standard storage pool.

Description

Table 97. Parameter differences for child pools and storage pools

Parameter	Child pool usage	Storage pool usage
-name	Optional	Optional for both parent pools and child pools
-mdisk	Cannot be used with child pools.	Optional
-tier	Cannot be used with child pools.	Optional
-easytier	Cannot be used with child pools.	Optional
-size	Mandatory	Cannot be used with parent pools.
-parentmdiskgrp	Mandatory	Cannot be used with parent pools.
-ext	Cannot be used for child pools.	Mandatory
-unit	Optional	Optional for both parent pools and child pools
-warning	Optional	Optional for both parent pools and child pools
-encrypt	Optional	Optional for both parent pools and child pools
-datareduction	Cannot be used with **-parentmdiskgrp** **Note:** A child pool cannot be created from a data reduction pool.	Data reduction pools must be a parent pool.

The **mkmdiskgrp** command creates a new storage pool and assigns the storage pool name if specified. The ID of the new storage pool is returned if the command is successful. Storage pools are collections of managed disks. Each storage pool is divided into chunks, called extents, which are used to create volumes.

Optionally, you can specify a list of managed disks that are added to this storage pool. These managed disks cannot belong to another storage pool, and they must have a mode of unmanaged. Use the **lsmdiskcandidate** command to get a list of suitable candidates. If **-tier** is specified, it applies to all of the MDisks.

Each managed disk that is a member of this group is split into extents. The storage that is available on these disks is added to a pool of extents that is available in this group. When a volume is created from this group, free extents from the pool are used, in accordance with the policy used when the volume was first created.

Subsequently, all managed disks added to this group are split into extents of the same size as the size that is assigned to the group.

When you choose an extent size, be aware of the amount of storage you want to virtualize in this group. The system maintains a mapping of extents between volumes and managed disks. The clustered system (system) can only manage a finite number of extents (4 194 304). One system can virtualize the following number of extents:

- 64 TB – if all storage pools have extent sizes of 16 MB.
- 2 PB – if all storage pools have extent sizes of 512 MB.
- 32 PB – if all storage pools have extent sizes of 8192 MB.

Important: The extent size for the storage pool can also limit volume size. Consider the maximum volume size that you want to use when you create storage pools. Refer to the information on creating storage pools for a comparison of the maximum volume capacity for each extent size. The maximum is different for thin-provisioned volumes.

Note: When an image mode volume is created, the storage pool increases in capacity by the size of the image mode volume (not the MDisk capacity) because the image mode volume might be smaller than the MDisk itself. If an extent is migrated from the image mode volume or MDisk to elsewhere in the group, the volume becomes a striped volume (no longer image mode). At this point, the available capacity might increase because the extra capacity available on the MDisk (for example, the capacity that was not part of the image mode volume) becomes available.

When you specify `-name` *`pool_name`* if you do not also specify **`-parentmdiskgrp`**, you create a parent pool where *pool_name* is the name of the new storage pool. When you specify `-name` *`pool_name`*, if you also specify **`-parentmdiskgrp`** and a size for it, you create a child pool where *pool_name* is the name of the new storage pool.

Note:

A data reduction pool is created by using the **`-datareduction`** parameter set to `yes`. The pool can be used to create fully allocated, thin or compressed volumes, or volume copies.

There is a maximum number of four data reduction pools in a system. When this limit is reached, creating any further pools with **`-datareduction`** set to `yes` is not possible.

The **`-datareduction`** parameter cannot be used to create a child pool. Creating a child pool whose parent is already a data reduction pool is not allowed.

An invocation example

This example adds a list of MDisks to the storage pool.

```
mkmdiskgrp -mdisk mdisk0:mdisk1:mdisk2:mdisk3 -ext 32
```

The resulting output:

```
MDisk Group, id [0], successfully created
```

An invocation example

This example specifies tier and Easy Tier information when you add a list of MDisks to the storage pool.

```
mkmdiskgrp -mdisk mdisk13:mdisk16 -ext 512 -tier tier_nearline -easytier measure
```

The resulting output:

```
MDisk Group, id [13], successfully created
```

An invocation example

This example creates a child pool from a parent pool.

```
mkmdiskgrp -size 100 -unit tb -parentmdiskgrp phypool
```

The resulting output:

```
MDisk Group, id [3], successfully created
```

An invocation example

This example creates a child pool from a parent pool and specifies an owner type.

```
mkmdiskgrp -parentmdiskgrp p0 -size 100 -unit gb -owner vvol_child_pool
```

The resulting output:

```
MDisk Group, id [3], successfully created
```

An invocation example

This example creates an encrypted child pool from a parent pool.

```
mkmdiskgrp -parentmdiskgrp 2 -name _my_encrypted_child_pool -encrypt yes -size 10 -unit gb
```

The resulting output:

```
MDisk Group, id [5], successfully created
```

An invocation example

This example creates an empty data reduction pool.

```
mkmdiskgrp -ext 512 -datareduction yes
```

The resulting output:

```
MDisk Group, id [16], successfully created
```

An invocation example

This example creates a data reduction pool with MDisks.

```
mkmdiskgrp -ext 512 -mdisk 3:5:6 -datareduction yes
```

The resulting output:

```
MDisk Group, id [17], successfully created
```

rmmdisk

Use the **rmmdisk** command to delete a managed disk (MDisk) from a storage pool.

Syntax

```
►►── rmmdisk ── ── -mdisk ──┬─ mdisk_id_list ───┬── ─┬──────────┬── ──┬─ mdisk_group_id ───┬──────────►◄
                            └─ mdisk_name_list ─┘    └─ -force ─┘     └─ mdisk_group_name ─┘
```

Parameters

-mdisk *mdisk_id_list* | *mdisk_name_list*
: (Required) Specifies one or more managed disk IDs or names to delete from the group.

-force
: (Optional) Migrates data on the specified disks to other disks in the group. The command completes asynchronously if **-force** is specified.

mdisk_group_id | *mdisk_group_name*
: (Required) Specifies the ID or name of the storage pool to delete the disks from. The warning threshold for a storage pool is automatically scaled when MDisks are deleted.

Description

This command attempts to remove the managed disk or disks from the group.

Remember: This command cannot be used for child pools.

Deleting a managed disk from a group can be done only if the managed disk does not contain any extents in use by a volume. If there are extents in use and you do not supply the force flag, the command fails.

Attention: If this disk being removed has already been powered down, removed, or is experiencing a power outage, the migration is pending and does not complete until the MDisk comes back online. The MDisk is not removed from the list of MDisks that are contained in the group.

If the disk has been deliberately removed, the only method of removing the MDisk is to remove the entire group itself.

Ensure that you do not destroy any controller LUNs until you delete them from the storage pool that they belong to.

The **rmmdisk** command fails if there are insufficient free extents on other disks in the storage pool for the duration of the command.

If you do specify the force flag, an attempt is made to migrate the extents that are in use onto other free extents within the storage pool. If there are not enough free extents in the storage pool, the command will fail even if the force flag is specified.

When an array MDisk is in a storage pool, five extents in the storage pool are reserved for internal use. If you attempt to remove an MDisk when an array MDisk is in the storage pool, the command fails (even if the **-force** flag is specified), if five free extents do not remain in the storage pool.

To delete the disks from the group, you have the following options:

- You can delete the volume that is using the extents that are specified on the managed disk.
- You can add more managed disks to the group, rerun the command and specify the **-force** parameter.

When data is being migrated from the managed disk, it might take some time for the command to complete. The command itself returns with a success code, notifying you that migration is in progress (if migration is required). An event is logged when the migration is complete and the disk is deleted from the group at this time. You can also check the progress of any active migrations by running the **lsmigrate** command.

If you specify **-force**, the **rmmdisk** command fails if there are offline MDisks. If there are no online quorum disks the migration fails.

Remember: When using the **-mdisk** parameter, MDisks are removed if there is one (or more) SAS MDisk specified in the list.

An invocation example

```
rmmdisk -mdisk mdisk12 -force Group3
```

The resulting output:

```
No feedback
```

rmmdiskgrp

Use the **rmmdiskgrp** command to delete a storage pool without being able to recover it.

Syntax

```
►►── rmmdiskgrp ── ─┬────────┬─ ─┬─ mdisk_group_id ───┬──────────────────────────►◄
                    └─ -force ─┘  └─ mdisk_group_name ─┘
```

Parameters

-force
(Optional) Specifies that all volumes and host mappings be deleted. When you use this parameter, all managed disks in the storage pool are removed and the storage pool itself is deleted.

Remember:

- You must specify **-force** to delete a child pool if it contains volume.
- You cannot specify **-force** to delete a parent pool if it has child pools.

Note: The command fails if **-force** is used to delete an MDisk group if:

- Any of the VDisks in the MDisk group are mirrored across multiple MDisk groups (other than the one that is being deleted).
- AND any of the VDisk mirrors are out of sync.
- AND an attempt is made to delete the in-sync copy. Deleting the only in-sync copy requires **-force**. Otherwise, it isn't needed if the VDisk has another in-sync copy.
- AND the out-of-sync copy is a thin-provisioned or compressed copy in a data reduction pool.

mdisk_group_id | *mdisk_group_name*
(Required) Specifies the ID or name of the storage pool that is to be deleted.

Note: You cannot delete a parent pool that has child pools. You must first delete the child pools.

Description

Important: Before you issue the command, ensure that you want to delete all mapping information. Data that is contained on the volume cannot be recovered after the storage pool is deleted.

The **rmmdiskgrp** command deletes the specified storage pool. The **-force** parameter is required if there are volumes that have been created from this storage pool or if there are managed disks in the storage pool. Otherwise, the command fails.

Note: This command also removes any associated storage pool throttling.

Deleting a storage pool is essentially the same as deleting a clustered system (system) or part of a system because the storage pool is the central point of control of virtualization. Because volumes are created by using available extents in the storage pool, mapping between volume extents and managed disk extents is controlled based on the storage pool.

The command deletes all volume copies in the specified storage pool. If the volume has no remaining synchronized copies in other storage pools, the volume is also deleted.

This command deletes the associated MDisk group (storage pool) throttle if that storage pool is removed.

Remember: This command is unsuccessful if:

- Volume protection is enabled (by using the **chsystem** command).
- The MDisk being removed is mapped to any volume that received I/O within the defined volume protection time period.

Remember: This command partially completes asynchronously. All volumes, host mappings, and Copy Services relationships are deleted before the command completes. The deletion of the storage pool then completes asynchronously.

In detail, if you specify the **-force** parameter and the volumes are still using extents in this storage pool, the following actions are initiated or occur:

- The mappings between that disk and any host objects and the associated Copy Services relationships are deleted.
- If the volume is a part of a FlashCopy mapping, the mapping is deleted.

  Note: If the mapping is not in the idle_or_copied or stopped states, the mapping is force-stopped and then deleted. Force-stopping the mapping might cause other FlashCopy mappings in the system to also be stopped. For more information, see the description for the **-force** parameter in the **stopfcmap** command.
- Any volume that is in the process of being migrated into or out of the storage pool is deleted. It frees up any extents that the volume was using in another storage pool.
- Volumes are deleted without first flushing the cache. Therefore, the storage controller LUNs that underlie any image mode MDisks might not contain the same data as the image mode volume before the deletion.
- If managed disks exist in the storage pool, all disks are deleted from the storage pool. They are returned to the unmanaged state.
- The storage pool is deleted.

Attention: If you use the **-force** parameter to delete all the storage pools in your system, you are returned to the processing state where you were after you added nodes to the system. All data that is contained on the volumes is lost and cannot be recovered.

An invocation example

```
rmmdiskgrp -force Group3
```

The resulting output:

```
No feedback
```

Chapter 26. User management commands

Use the user management commands to configure remote authentication service and manage users and groups on the clustered system.

chauthservice

Use the **chauthservice** command to configure the remote authentication service of the clustered system (system).

Syntax

```
►►── chauthservice ── ─┬────────────────────┬─ ─┬────────────────┬─ ─┬──────────────┬─ ─────────►
                       └─ -enable ──┬─ yes ─┬┘  └─ -type ── ldap ─┘  └─ -url ── url ─┘
                                    └─ no ──┘

►─┬──────────────────────────┬─ ─┬───────────────────────────────┬─ ─┬────────────────────────┬──►
  └─ -username ── user_name ─┘   └─ -password ─┬──────────────┬─┘   └─ -sslcert ── file_name ─┘
                                               └─ password ───┘

►─ ─┬─────────────┬────────────────────────────────────────────────────────────────────────────►◄
    └─ -refresh ──┘
```

Parameters

-enable yes | no
(Optional) Enables or disables the system's use of the remote authentication server. When the **enable** parameter is set to `no`, remote authentications are failed by the system, but local authentications continue to operate normally.

-type ldap
(Optional) Specifies the authentication service type (which must be LDAP). An LDAP server must be configured.

Remember: The remote authentication service must be enabled (**-enable yes**) for this setting to come into effect.

-url *url*
(Optional - IBM Security Services only) Specifies the website address (URL) of Security Services, which is referred to as `TIP` in the CLI. The host part of the URL must be a valid numeric IPv4 or IPv6 network address. You can use the following characters in the URL:

- a - z
- A - Z
- 0 - 9
- _
- ~
- :
- [
-]
- %
- /

The maximum length of the URL is 100 characters.

This option is no longer used.

-username *user_name*
(Optional) Specifies the HTTP basic authentication user name. The user name cannot start or end with a blank. The user name can consist of a string of 1 - 64 ASCII characters except for the following characters:

- %
- :
- "
- ,
- *
- '

-password *password*
(Optional) Specifies the HTTP basic authentication user password. The password cannot start or end with a blank. It must consist of a string of 6 - 64 printable ASCII characters. The *password* variable is optional. If you do not provide a password, the system prompts you and does not display the password that you type.

-sslcert *file_name*
(Optional) Specifies the name of the file that contains the SSL certificate, in privacy enhanced mail (PEM) format, for the remote authentication server. The certificate file must be in valid PEM format and have a maximum length of 12 KB.

-refresh
(Optional) Causes the system to invalidate any remote user authorizations that are cached on the system. Use this option when you modify user groups on the authentication service and want the change to immediately take effect on the system.

Note: If you clear the cache, anyone who uses the system might have to log in again (for example, if credentials are provided to one of the defined LDAP servers).

Description

The system authenticates remote users by using Lightweight Directory Access Protocol (LDAP).

Before you enable remote authentication, ensure that the properties of the service are properly configured on the system. It is not necessary to disable the remote authentication service to change its properties. LDAP authentication can be configured by using the **chldap** command, and LDAP servers can be added to the system by using the **mkldapserver** command.

Remember: For the authentication type to be set to LDAP with authorization enabled (true), an LDAP server must be configured.

When the authentication service is enabled, the system does not test whether the remote authentication system is operating correctly.

- To establish whether the system is operating correctly, enter the **lscurrentuser** command for a remotely authenticated user. If the output lists the user roles that are obtained from the remote authentication server, remote authentication is operating successfully. If the output is an error message, remote authentication is not working correctly, and the error message describes the problem.
- To establish whether LDAP is operating correctly, in addition to the **lscurrentuser** command, enter the **testldapserver** command. The **testldapserver** command can be entered whether or not remote authentication is enabled, and can be used to test the connection to LDAP servers, as well as user authorization and authentication.

To disable the remote authentication service in a controlled manner when it is not available, use the **enable** parameter with the no option.

An invocation example

To disable remote authentication, enter the following command:

```
chauthservice -enable no
```

The following text is displayed when the command runs:

```
No feedback
```

An invocation example

To refresh the system remote authorization cache, enter the following command:

```
chauthservice -refresh
```

The following text is displayed when the command runs:

```
No feedback
```

chcurrentuser

Use the **chcurrentuser** command to change the attributes of the current user.

Syntax

```
►►─ chcurrentuser ─ ┬───────────────────────────────────────┬─┬──────────────────────────────┬──►◄
                    ├─ -password ─┬────────────────────────┬┤ ├─ -keyfile ─ sshkey_filename ─┤
                    │             └─ cleartext_password ───┘│ └─ -nokey ────────────────────┘
                    └─ -nopassword ─────────────────────────┘
```

Parameters

-password *cleartext_password*
(Optional) Specifies the new password to be associated with the current user. The password cannot start or end with a blank. It must consist of a string of 6 - 64 printable ASCII characters. You can optionally specify the password with the **password** parameter. If you do not specify the password, the system prompts you for it before running the command and does not display the password that you type. Either the **password** parameter or the **nopassword** parameter can be set.

-nopassword
(Optional) Specifies that the user's password is to be deleted.

-keyfile *sshkey_filename*
(Optional) Specifies the name of the file that contains the Secure Shell (SSH) public key. Either the **keyfile** parameter or the **nokey** parameter can be set.

-nokey
(Optional) Specifies that the user's SSH key is to be deleted.

Description

Use the **chcurrent user** command to modify the attributes of the current user.

An invocation example

```
chcurrentuser -password secret -nokey
```

The resulting output:

```
No feedback
```

chldap

Use the **chldap** command to change system-wide Lightweight Directory Access Protocol (LDAP) configuration. This command can be used to configure remote authentication with LDAP. These settings apply when authenticating against any of the LDAP servers configured using the **mkldapserver** command.

Syntax

```
►►── chldap── ┬──────────────────────────┬──────────────────────────────────────────►
              ├─ -type ──┬─ ad ────┬─────┤
              │          ├─ itds ──┤     │
              │          └─ other ─┘     │
              └─ -reset ─────────────────┘

►─┬──────────────────────────────────────────────────────────┬─┬──────────────────────────┬──►
  └─ -username ── username ──┬─────────────────────────────┬─┘ └─ -security ──┬─ tls ──┬─┘
                             ├─ -password ──┬────────────┬─┤                  ├─ ssl ──┤
                             │              └─ password ─┘ │                  └─ none ─┘
                             └─ -encpassword ──┬────────────┬┘
                                               └─ password ─┘

►─┬──────────────────────────────────────┬─┬─────────────────────────────────────────┬──►
  └─ -userattribute ── user_attribute ───┘ └─ -groupattribute ── group_attribute ────┘

►─┬───────────────────────────────────────────────┬─┬───────────────────────────────────────────┬──►◄
  └─ -auditlogattribute ── auditlogattribute ─────┘ └─ -nestedgroupsearch ──┬─ client ─┬────────┘
                                                                            ├─ server ─┤
                                                                            └─ off ────┘
```

Parameters

-type *ad* |*itds*|*other* | **-reset**
(Optional) Specify the LDAP server type, or reset LDAP configuration to defaults for the current server type. Defaults for the configured server type:

- Active Directory (AD)
- IBM Security Directory Server (ISDS)
- Other

-username *username*
(Optional) Specifies a username for administrative binding. This can be:

Note:

- A distinguished name (DN)
- A user principal name (UPN) or NT login name for Active Directory

-password *password*
(Optional) Specifies the password for the administrative binding. You can optionally specify the

password with this parameter. If you do not specify the password, the system prompts you for it before running the command and does not display the password that you type.

-encpassword *password*
: (Optional) Specifies the password for the enclosure. You can optionally specify the password with this parameter. If you do not specify the password, the system prompts you for it before running the command and does not display the password that you type.

-security *tls* | *ssl* | *none*
: (Optional) Specifies the type of security to use when communicating with LDAP servers. Specifying `tls` enables Transport Layer Security (TLS) security. Specifying `ssl` enables Secure Socket Layer (SSL) security. The default value is `none`.

-userattribute *user_attribute*
: (Optional) Specifies the LDAP attribute used to determine the user name of remote users. The user attribute must exist in your LDAP schema and must be unique for each of your users.

-groupattribute *group_attribute*
: (Optional) Specifies the LDAP attribute used to determine the group memberships of remote users. The attribute must contain either the DN of a group or a colon-separated list of group names.

-auditlogattribute *auditlogattribute*
: (Optional) Specifies the LDAP attribute used to determine the identity of remote users. When a user performs an audited action, this information is recorded in the audit.

-authcacheminutes *auth_cache_minutes*
: (Optional) Specifies the period for which to cache authentication details.

-nestedgroupsearch *client* | *server* | *off*
: (Optional) Specifies whether nested groups are evaluated on the client (clustered system), server (authentication service), or are not evaluated not at all.

Description

At least one parameter must be specified.

The **chldap** command can be run whether or not LDAP authentication is enabled. Specifying **-reset** or **-type** populates the default values unless otherwise specified.

You can only specify **-password** or **-encpassword** if **-username** is specified.

The **-type** parameter values are only set to defaults for the specified type if the type is different from the existing type.

If the type is **itds**, **-nestedgroupsearch** cannot be executed (nested groups are evaluated by default). If the type is **ad**, **-nestedgroupsearch** can only be set to **client** or **off** because there is no server support. If the type is **other**, the **-nestedgroupsearch** parameter is fully configurable.

Use **-username** to specify a distinguished name (DN), user principal name (UPN), or NT login name. Distinguished names (DN) must be a sequence of attribute=value pairs separated by a comma (`,`), semi-colon(`;`), or plus sign (+). A backslash (`\,`) must be used to escape special characters, and can also be used to specify UTF-8 characters using their byte encoding. For example, `c` acute can be represented as `\C4\87`. NT logins are valid for only the Active Directory and must be in the `DOMAIN\user` format. These logins must not start or end with a period (`.`) and both the DOMAIN and the user must not use the following characters: `\/:?"<>|` UPN logins are valid for Active Directory only and must be in the format user@suffix. Both user and suffix can not use spaces or the following characters: `()<>,;:\"[]@`

Tip:

- Remember that **-userattribute**, **-groupattribute**, and **-auditlogattribute** accept values that:

1. Must begin with a letter
2. Only contain ASCII letters, digit characters, and hyphens
3. Are case-insensitive

The following LDAP (first-time) configuration suggestions assist with LDAP server setup:

Important:

- Ensure that the system is configured appropriately according to your LDAP schema. Issue **`chldap -type`** to populate the system's LDAP configuration with the server type defaults. Issue **`chldap -reset`** to return to these defaults at any time.
  - (Advanced) For all server types, users are authenticated with a username configured in the LDAP attribute `user_attribute`. This attribute must exist in the LDAP schema and must be unique for each user. It is configurable by issuing **`chldap -userattribute`**. Active Directory users can also authenticate using their UPN or NT login names.
  - (Advanced) Authenticated users are assigned roles according to their LDAP group memberships. Each user's group memberships must be stored in the LDAP attribute `group_attribute`. This can be either an LDAP attribute containing the DN of the user's LDAP group, or an LDAP attribute containing a colon-separated list of user group names. It is configurable by issuing **`chldap -groupattribute`**.
  - (Advanced) When an LDAP authenticated user runs a command that is audited, the user's login name is placed in the audit log. The name is extracted from the LDAP attribute `audit_log_attribute`, which is configurable by issuing **`chldap -auditlogattribute`**.
- Ensure that the system is able to search within the user and group trees on LDAP servers. By default the system authenticates anonymously. Consequently, you must either permit anonymous searches of the LDAP directory, or create an LDAP user with the appropriate permissions and issue the **`chldap -username`** and **`chldap -password`** commands to instruct the system to search as this user.
- Ensure that the system is able to connect with the appropriate level of security. Passwords are sent to the LDAP server as clear text, so Transport Layer Security (TLS) encryption is recommended. Issue **`chldap -security`** to change the security level.
- (Advanced): On Active Directory and some other LDAP servers, the system (by default) identifies groups to which users belong directly. To assign users permissions according to a parent group, enable the nested group search on the client by issuing **`chldap -nestedgroupsearch`**. This setting has an additional performance overhead and supports up to 8 levels of nesting.

An invocation example

```
chldap -type
itds -username uid=joebloggs,cn=admins,dc=company,dc=com -password passw0rd
-auditlogattribute descriptiveName
```

The resulting output:

```
No feedback
```

chldapserver

Use the **`chldapserver`** command to modify a Lightweight Directory Access Protocol (LDAP) server.

Syntax

```
►►— chldapserver — ┬───────────────────┬─┬───────────────────────┬─┬─────────────────┬──────►◄
                   └─ -ip — ip_address ─┘ └─ -name — server_name ─┘ └─ -port — port ─┘
```

```
►──┬──────────────────────────┬──┬──────────────────────┬──┬──────────────────────┬──────►
   ├─ -sslcert ── file_name ──┤  ├─ -basedn ── base_dn ─┤  └─ -preferred ──┬─ yes ─┬─┘
   └─ -nosslcert ─────────────┘  └─ -nobasedn ──────────┘                   └─ no ──┘

►──┬─ ldap_server_id ───┬──────────────────────────────────────────────────────────────►◄
   └─ ldap_server_name ─┘
```

Parameters

-ip *ip_address*
(Optional) Specifies the server IP address (Internet Protocol Version 4 or 6).

-name *server_name*
(Optional) Specifies the LDAP server name.

-port *port*
(Optional) Specifies the LDAP server port. The default value (if you do not specify a value) is 389. If you specify TLS security the value is 389 and if you specify SSL security the value is 636.

-sslcert *file_name* | **-nosslcert**
(Optional) Set (**-sslcert**) or clear (**-nosslcert**) the secure socket layer (SSL) certificate.

-basedn *base_dn* | **-nobasedn**
(Optional) Use the base distinguished name (DN) for search (**-nobasedn** indicates to use the default DN).

-preferred *yes* | *no*
(Optional) Specifies whether the server is preferred over other configured LDAP servers (or not preferred).

ldap_server_id | *ldap_server_name*
(Required) Specifies the LDAP server ID or name.

Description

Important: During normal operation, LDAP requests are sent to **-preferred** servers depending on availability. If no servers are marked as **-preferred**, LDAP requests are sent to configured servers based on availability.

If **-sslcert** is specified, the server certificate is verified while authenticating. The SSL certificate must exist on the current node. If **-nosslcert** is specified, any certificate file is deleted and the server certificate is not checked.

The **-basedn** parameter indicates the distinguished name (DN) to use as a base from which to search for users in the LDAP directory. If Transport Layer Security (TLS) is enabled and **-sslcert** is specified, the server certificate is verified during authentication. The secure socket layer (SSL) certificate must exist on the node being used. Otherwise, a server certificate is not checked.

The clustered system (system) must be configured with an appropriate version IP address when **-ip** is specified. The IP address specified with the **-ip** parameter must be of a version supported by the system. The certificate file must be in valid PEM format and have a maximum length of 12 kilobytes.

Distinguished names must be a sequence of attribute=value pairs separated by a comma (**,**), semi-colon(**;**), or plus sign (+) escaping special characters with \ where appropriate, and specified UTF-8 characters using their byte encoding. For example, **,** for commas or \C4\87 for the UTF-8 character c acute.

This command runs whether or not LDAP authentication is enabled.

Remember: There can be a maximum of six configured LDAP servers. If you attempt to create a seventh LDAP server an error is returned.

An invocation example with basic server details

```
chldapserver -ip 192.135.60.3 -port 400 ldapserver0
```

The resulting output:

```
No feedback
```

An invocation example specifying an SSL certificate

```
chldapserver -sslcert /tmp/activedirectorycert.pem 0
```

The resulting output:

```
No feedback
```

An invocation example to remove an SSL certificate

```
chldapserver -nosslcert 0
```

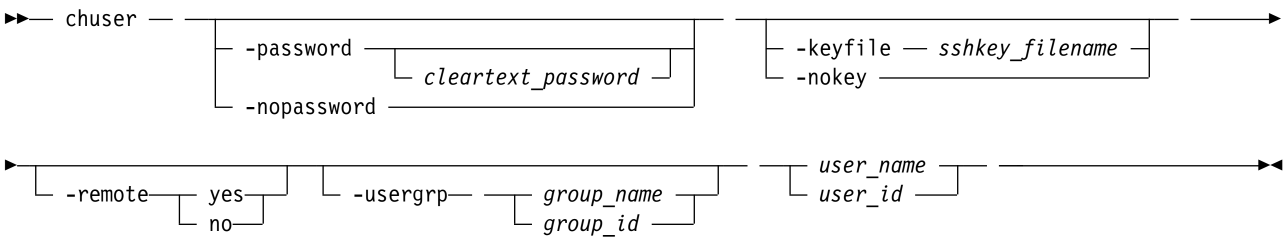
The resulting output:

```
No feedback
```

chuser

Use the **chuser** command to change the attributes of an existing user.

Syntax



Parameters

-password *cleartext_password*
(Optional) Specifies the new password to be associated with the user. The password cannot start or end with a blank. It must consist of a string of 6 - 64 printable ASCII characters. You can optionally specify the password with the **password** parameter. If you do not specify the password, the system prompts you for it before running the command and does not display the password that you type. Either the **password** parameter or the **nopassword** parameter can be set.

-nopassword
(Optional) Specifies that the user's password is to be deleted.

-keyfile *sshkey_filename*
(Optional) Specifies the name of the file that contains the Secure Shell (SSH) public key. Either the **keyfile** parameter or the **nokey** parameter can be set.

-nokey
(Optional) Specifies that the user's SSH key is to be deleted.

-remote yes | no
: (Optional) Specifies whether the user authenticates to the cluster using a remote authentication service. Either yes or no must be set.

-usergrp *group_name | group_id*
: (Optional) Specifies the new group for the user.

user_name | user_id
: (Required) Specifies the user whose attributes are to be changed.

Description

Use the **chuser** command to modify the attributes of an existing user.

You must have the Security Administrator role to create, delete, or change a user.

Only use the **usergrp** parameter for local users. If you change a user from local to remote, the user's association with any group is removed.

If you change a user from remote to local, a user group must be specified. If you change a user from local to remote, the user must have both a password and an SSH key.

If you use the **keyfile** parameter, the SSH key file should be placed in the /tmp directory before running this command. When you run the command, the SSH key is copied into cluster state and activated for the user, and the input file is deleted.

An invocation example

```
chuser -remote no -usergrp Monitor -nokey jane
```

The resulting output:

```
No feedback
```

chusergrp

Use the **chusergrp** command to change the attributes of an existing user group.

Syntax

```
►►— chusergrp — — -role ——┬— role_id ——┬┬————————————————————┬— ——┬group_id——┬——————————————————►◄
                          └— role_name —┘└— -remote——┬yes—┬┘     └group_name┘
                                                     └no——┘
```

Parameters

-role *role_name*
: (Optional) Specifies the role to be associated with users that belong to this group. One of the following roles must be selected: Monitor, CopyOperator, Service, Administrator, or SecurityAdmin.

-remote yes | no
: (Optional) Specifies whether this user group should be used to set the role of remote users. Either the yes or no option must be set.

group_id | group_name
: (Required) The ID or name of the user group whose attributes are to be changed.

Description

Use the **chusergrp** command to modify the attributes of an existing user group.

You must have the Security Administrator role to create, delete, or change a user.

The roles of the default groups cannot be changed.

An invocation example

```
chusergrp -role Administrator admin
```

The resulting output:

```
No feedback
```

lscurrentuser

Use the **lscurrentuser** command to display the name and role of the logged-in user.

Syntax

```
►►── lscurrentuser ── ─┬─────────┬─ ─┬────────────────────┬──────────────────────────►◄
                       └─ -nohdr ─┘   └─ -delim ── delimiter ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. If you enter `-delim :` on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

Description

This command displays the name and role of the current user.

An invocation example

```
lscurrentuser
```

The resulting output:

```
name superuser role_name SecurityAdmin
```

lsldap

Use the **lsldap** command to display the details for the system-wide Lightweight Directory Access Protocol (LDAP) configuration.

Syntax

```
►►── lsldap ── ─┬─────────┬─ ─┬──────────────────────┬──────────────────────────────►◄
                └─ -nohdr ┘   └─ -delim ── delimiter ┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default, in a concise view all columns of data are space-separated, with the width of each column set to the maximum width of each item of data. In a detailed view, each item of data is an individual row, and if you display headers, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. Enter `-delim :` on the command line, and the colon character (`:`) separates all items of data in a concise view (for example, the spacing of columns does not occur); in a detailed view, the specified *delimiter* separates the data from its header.

Description

Table 98 provides the attribute values that can be displayed as output view data.

Table 98. lsldap attribute values

Attribute	**Value**
type	Indicates the LDAP server type. The values are: • `ad` indicates that it is an Active Directory server. • `itds` indicates that it is an IBM Tivoli Directory Server. • `other` indicates that it is another type of server.
enabled	Indicates whether native LDAP authentication is enabled. The value is `yes` or `no/`
error_sequence_number	Indicates the sequence number of non-fixed LDAP configuration error log. The value is a number (integer).
username	Indicates the binding user name or distinguished name. The value is an alphanumeric string or blank if there is no name.
security	Indicates the type of security in use. The values are: • `tls` indicates that it is Transport Layer Security. • `none` indicates that there is no security.
user_attribute	Indicates the LDAP attribute that represents the user login.
group_attribute	Indicates the LDAP attribute that represents the user group membership.
audit_log_attribute	Indicates the LDAP attribute that represents the user name in audit log.
auth_cache_minutes	Indicates the period (in minutes) for which to cache session details.
nested_group_search	Indicates the handling of nested groups. The values are: • `off` indicates that there is no nested group handling search. • `client` indicates that the system must search for nested groups on the client. • `server` indicates that the system must search for nested groups on the server.

An invocation example

```
lsldap -delim :
```

The resulting output:

```
type:ad
enabled:yes
error_sequence_number:12
username:admin@company.com
security:tls
user_attribute:sAMAccountName
group_attribute:memberOf
audit_log_attribute:userPrincipalName
auth_cache_minutes:10
nested_group_search:off
```

lsldapserver

Use the **lsldapserver** command to display the most recent details for all configured Lightweight Directory Access Protocol (LDAP) servers.

Syntax

```
►►— lsldapserver — ┬────────────┬─ ┬──────────────────────┬─ ┬────────────────────┬──────────►◄
                   └─ -nohdr ───┘  └─ -delim — delimiter ─┘  ├─ ldap_server_id ───┤
                                                             └─ ldap_server_name ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default, in a concise view all columns of data are space-separated, with the width of each column set to the maximum width of each item of data. In a detailed view, each item of data is an individual row, and if you display headers, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. Enter `-delim :` on the command line, and the colon character (`:`) separates all items of data in a concise view (for example, the spacing of columns does not occur); in a detailed view, the specified *delimiter* separates the data from its header.

ldap_server_id | *ldap_server_name*
(Optional) Specifies the ID or name for LDAP server that is being used.

Description

Remember:

- The base distinguished name (DN) is at the end of the concise view information; other fields must be added before the base DN.
- The command fails if a server is specified that does not exist.

Table 99 on page 627 provides the attribute values that can be displayed as output view data.

Table 99. lsldapserver attribute values

Attribute	Value
id	Specifies the ID of the LDAP server.
name	Specifies the name of the LDAP server.
error_sequence_number	Specifies the sequence number of non-fixed LDAP server error log.
IP_address	Specifies the IP address of the LDAP server (Internet Protocol Versions 4 and 6).
port	Specifies the LDAP server port. The default value is 389. The value for TLS security is 389 and the value for SSL security is 636.
cert_set	Specifies the certificate setting if a certificate is configured.
preferred	Specifies the server preference (preferred server).
base_dn	Specifies the base distinguished name (DN) that is used in LDAP searches.

Description

This command displays details for the configured LDAP servers.

Note: There is a maximum of six configured LDAP servers.

A concise invocation example

```
lsldapserver -delim :
```

The resulting output:

```
id:name:error_sequence_number:IP_address:port:cert_set:preferred:base_dn
0:ldapserver0::192.135.60.3:389:no:yes:ou=users,dc=company,dc=com
1:ldapserver1:12:192.135.60.4:389:no:no:ou=users,dc=company,dc=com
2:ldapserver2::192.135.60.5:389:yes:yes:ou=users,dc=company,dc=com
3:ldapserver3::192.135.60.6:389:yes:no:ou=users,dc=company,dc=com
```

A detailed invocation example

```
lsldapserver -delim : ldapserver0
```

The resulting output:

```
id:0
name:ldapserver0
error_sequence_number:
IP_address:192.135.60.3
port:389
cert_set:no
preferred:yes
base_dn:ou=users,dc=company,dc=com
```

lsuser

Use the **lsuser** command to display a list of the users that is created on the clustered system (system).

Syntax

```
►►— lsuser — ┬──────────┬— ┬─────────────────────┬— ─────────────────────────────►
             └─ -nohdr ─┘  └─ -delim — delimiter ─┘
```

```
►─┬────────────────────────────────────┬─ ┬───────────────────┬─ ┬────────────────┬──────────────►◄
  └─ -filtervalue ── attribute=value ──┘  └─ -filtervalue? ───┘  ├─ usergrp_name ─┤
                                                                   └─ usergrp_id ───┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. If you enter `-delim :` on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards with the SAN Volume Controller CLI:

- The wildcard character is the asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, enclose the filter entry within double quotation marks (""), as follows:

```
lsuser -filtervalue "usergrp_name=md*"
```

-filtervalue?
(Optional) Displays the valid filter attributes for the **-filtervalue***attribute=value* parameter:

- password
- ssh_key
- remote
- usergrp_id
- usergrp_name

usergrp_name | *usergrp_id*
(Optional) Specifies the ID or name of the user for which the association is being deleted. If this parameter is specified, the detailed view for the specified user is displayed in the output. If you do not specify an ID or name, the concise view is displayed.

Description

This command displays a list of users that is created on the system.

A concise invocation example

```
lsuser
```

The resulting output:

```
id   name        password   ssh_key   remote   usergrp_id   usergrp_name
0    superuser   yes        no        no       0            SecurityAdmin
1    simon       no         yes       no       2            CopyOperator
2    jane        yes        no        no       3            Service
3    kip         yes        yes       yes
```

A detailed invocation example

```
lsuser 1
```

The resulting output:

```
id 1
name tpc_admin
password yes
ssh_key no
remote no
usergrp_id 0
usergrp_name SecurityAdmin
```

lsusergrp

Use the **lsusergrp** command to display a list of the user groups that is created on the clustered system (system).

Syntax

```
>>-- lsusergrp -- +-----------+ +----------------------+ ------------------------->
                  '- -nohdr --' '- -delim -- delimiter-'

>-+---------------------------------+ +-----------------+ +-usergrp_name-+-------><
  '- -filtervalue -- attribute=value' '- -filtervalue? -' '-usergrp_id---'
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. If you enter `-delim :` on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards with the SAN Volume Controller CLI:

- The wildcard character is the asterisk (*), which must be used as the first or last character in the string.

- The command can contain a maximum of one wildcard.
- When you use a wildcard, enclose the filter entry within double quotation marks (""), as follows:
  `lsusergrp -filtervalue "role=md*"`

-filtervalue?
(Optional) Displays the valid filter attributes for the **-filtervalue** *attribute=value* parameter:
- `role_id`
- `role_name`
- `remote`

usergrp_name **|** *usergrp_id*
(Optional) Specifies the ID or name of the user group to view. If you do not specify an ID or name, all groups are displayed.

Description

This command displays a list of user groups that is created on the system.

An invocation example

```
lsusergrp
```

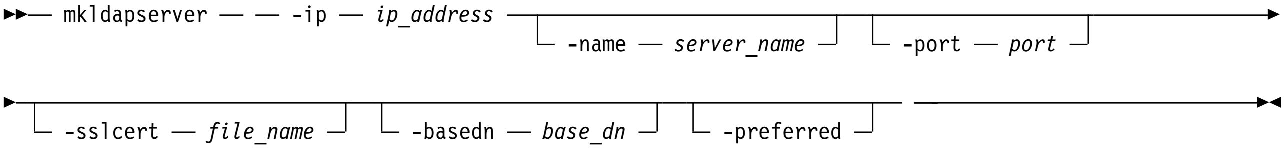
The resulting output:

```
id      name              role              remote
0       SecurityAdmin     SecurityAdmin     yes
1       Administrator     Administrator     no
2       CopyOperator      CopyOperator      no
3       Service           Service           yes
4       Monitor           Monitor           no
5       support           Service           no
```

mkldapserver

Use the **mkldapserver** command to display the data used to create a Lightweight Directory Access Protocol (LDAP) server.

Syntax



Parameters

-ip *ip_address*
(Required) Specifies the server IP address (Internet Protocol Version 4 or 6).

-name *server_name*
(Optional) Specifies the LDAP server name.

-port *port*
(Optional) Specifies the LDAP server port. The default value (if you do not specify a value) is 389. If you specify TLS security the value is 389 and if you specify SSL security the value is 636.

-sslcert *file_name*
(Optional) Set the SSL certificate.

-basedn *base_dn*
(Optional) Use the base distinguished name for search.

-preferred
(Optional) Specifies that this server is preferred over other configured LDAP servers.

Description

Important: During normal operation, LDAP requests are sent to **-preferred** servers depending on availability. If no servers are marked as **-preferred**, LDAP requests are sent to configured servers based on availability.

If **-sslcert** is specified, the server certificate is verified while authenticating.

Note: The SSL certificate must exist on the current node.

The **-basedn** parameter indicates the distinguished name (DN) to use as a base from which to search for users in the LDAP directory. If Transport Layer Security (TLS) is enabled and **-sslcert** is specified, the server certificate is verified during authentication. The secure socket layer (SSL) certificate must exist on the node being used, otherwise a server certificate is not checked.

The clustered system (system) must be configured with an appropriate version IP address when **-ip** is specified. The IP address specified with the **-ip** parameter must be of a version supported by the system. The certificate file must be in valid PEM format and have a maximum length of 12 kilobytes.

Distinguished names must be a sequence of `attribute=value` pairs separated by a comma (**,**), semi-colon(**;**), or plus sign (+) escaping special characters with a backslash (\) where appropriate, and specified UTF-8 characters using their byte encoding. For example, **\,** for commas or `\C4\87` for the UTF-8 character `c` acute.

This command runs whether or not LDAP authentication is enabled.

Remember: There is a maximum of six configured LDAP servers. Attempting to create a seventh LDAP server returns an error.

An invocation example

```
mkldapserver -ip 192.135.60.3
```

The resulting output:

```
LDAP Server, id [0], successfully created
```

mkuser

Use the **mkuser** command to create either a local or a remote user to access a clustered system (system).

Syntax

```
►►── mkuser ── ── -name ── user_name ──────────────────────────────────────►

►──┬─ -remote──────────────────────────────────────────────────┬───────────►◄
   └─ -usergrp── ──┬group_id───┬──┬─ -keyfile ── sshkey_filename ────────┤
                   └group_name─┘  └─ -password ──┬─────────────────────┬─┘
                                                 └─ cleartext_password ─┘
```

Parameters

-name *user_name*
(Required) Specifies the unique user name. The user name cannot start or end with a blank. The user name must consist of a string of 1 - 256 ASCII characters, with the exception of the following characters: %:",*' .

-remote | -usergrp
(Required) Specifies whether the user authenticates to the system using a remote authentication service or system authentication methods.Either the **remote** parameter or the **usergrp** parameter must be set. If **usergrp** is specified, it must be followed by *group_name* or *group_id* (see next parameter).

group_name **|** *group_id*
(Required if **usergrp** is specified) The ID or name of the user group with which the local user is to be associated.

-password *cleartext_password*
(Optional) Specifies the password to be associated with the user. The password cannot start or end with a blank. It must consist of a string of 6 - 64 printable ASCII characters. You can optionally specify the password with the **password** parameter. If you do not specify the password, the system prompts you for it before running the command and does not display the password that you type.

-keyfile *sshkey_filename*
(Optional) Specifies the name of the file that contains the Secure Shell (SSH) public key.

Description

The **mkuser** command creates a new local or remote user to access a system. The command returns the ID of the created user.

You must have the Security Administrator role to create, delete, or change a user.

If you create a local user, you must specify the existing user group that the user belongs to. All local users must have a group. The user group defines roles that provide the user with access to specific operations on the system. You must also specify either the **keyfile** or **password** parameter, or both.

If you create a remote user, you must specify both the **keyfile** and **password** parameters. Remote users have their groups defined by the remote authentication service.

Up to 400 users can be defined on the system. You can also create new users and assign keys to them.

If you use the **keyfile** parameter, the SSH key file should be placed in the /tmp directory before running this command. When you run the command, the SSH key is copied into system state and activated for the user, and the input file is deleted.

An invocation example

```
mkuser -name jane -usergrp Service -password secret
```

The resulting output:

```
User, id [1], successfully created
```

mkusergrp

Use the **mkusergrp** command to create a new user group.

Syntax

```
►►— mkusergrp — — -name — group_name — -role ─┬─ role_id ──┬─┬──────────┬─ ───────────────►◄
                                              └─ role_name ─┘ └─ -remote─┘
```

Parameters

-name *group_name*
(Required) Specifies the unique user group name. The group name cannot start or end with a blank. The group name must consist of a string of 1 - 64 ASCII characters, with the exception of the following characters: `%:",*'` .

-role *role_id* | *name*
(Required) Specifies the role (by ID or name) to be associated with all users that belong to this user group. One of the following roles must be selected:

- `Monitor`
- `CopyOperator`
- `Service`
- `Administrator`
- `SecurityAdmin`
- `VasaProvider`
- `AdministratorWithoutDelete`

.

-remote
(Optional) Specifies that this user group should be used to set the role of remote users. This is disabled by default.

Description

The **mkusergrp** command creates a new user group to organize users of the SAN Volume Controller clustered system by role. Use the **lsusergrp** command to view a list of user groups that have been created on the clustered system.

You must have the security administrator role (`SecurityAdmin` role name) to create, delete, or change a user group.

Each user group has one role that determines the role of users that belong to that group. Use the **role** parameter to specify one of the following roles for the user group:

Monitor
You can issue any information display command and, additionally, the following commands:

- **finderr**
- **dumperrlog**
- **dumpinternallog**
- **chcurrentuser**
- **ping**
- **svcconfig backup**

CopyOperator
You can issue the following commands:

- **prestartfcconsistgrp**
- **startfcconsistgrp**

- **stopfcconsistgrp**
- **chfcconsistgrp**
- **prestartfcmap**
- **startfcmap**
- **stopfcmap**
- **chfcmap**
- **startrcconsistgrp**
- **stoprcconsistgrp**
- **switchrcconsistgrp**
- **chrcconsistgrp**
- **startrcrelationship**
- **stoprcrelationship**
- **switchrcrelationship**
- **chrcrelationship**
- **chpartnership**

In addition, you can issue all of the commands allowed by the **Monitor** role.

Service

You can issue the following commands:

- **applysoftware**
- **setlocale**
- **addnode**
- **rmnode**
- **cherrstate**
- **writesernum**
- **detectmdisk**
- **includemdisk**
- **clearerrlog**
- **cleardumps**
- **settimezone**
- **stopsystem**
- **startstats**
- **stopstats**
- **settime**

In addition, you can issue all of the commands allowed by the **Monitor** role.

Administrator

You can issue any command other than:

- **chauthservice**
- **mkuser**
- **rmuser**
- **chuser**
- **mkusergrp**
- **rmusergrp**
- **chusergrp**
- **setpwdreset**

VASAProvider

The system uses this role to implement the VMware Virtual Volumes function. It provides a group with users that can be used by that software. You can issue any command other than:

- **chauthservice**
- **chldap**
- **chldapserver**
- **chsecurity**
- **chuser**
- **chusergrp**
- **mkldapserver**
- **mkuser**
- **mkusergrp**
- **rmldapserver**
- **rmuser**
- **rmusergro**
- **setpwdreset**

SecurityAdmin

You can issue all commands.

The command returns the ID of the created user group.

An invocation example

```
mkusergrp -name support -role Service
```

The resulting output:

```
User Group, id [5], successfully created
```

An invocation example

```
mkusergrp -role VasaProvider -name myVasaProvider
```

The resulting output:

```
User Group, id [5], successfully created
```

An invocation example

```
mkusergrp -role AdministratorWithoutDelete -name myAdministratorWithoutDelete
```

The resulting output:

```
User Group, id [5], successfully created
```

rmldapserver

Use the **rmldapserver** command to delete a Lightweight Directory Access Protocol (LDAP) server.

Syntax

```
►►— rmldapserver — ┬─ ldap_server_id ───┬─ ─────────────────────────────────────►◄
                   └─ ldap_server_name ─┘
```

Parameters

ldap_server_id **|** *ldap_server_name*
(Required) Specifies the LDAP server ID or name to delete.

Description

Remember:

- If remote authentication with LDAP is enabled, the final LDAP server cannot be deleted. To delete the final LDAP server disable LDAP authentication by specifying **chauthservice -enable no**.
- The **rmldapserver** command can be specified whether or not LDAP authentication is enabled.

An invocation example

```
rmldapserver ldapserver0
```

The resulting output:

```
No feedback
```

rmuser

Use the **rmuser** command to delete a user.

Syntax

```
►►— rmuser — ──┬user_id───┬──────────────────────────────────────────────────►◄
               └user_name─┘
```

Parameters

user_id **or** *user_name*
(Required) Specifies the user to be removed.

Description

Use the **rmuser** command to delete a user.

You must have the Security Administrator role to create, delete, or modify a user.

An invocation example

```
rmuser jane
```

The resulting output:

```
No feedback
```

rmusergrp

Use the **rmusergrp** command to delete a user group.

Syntax

```
►►─rmusergrp─ ──┬────────┬─┬group_id───┬──────────────────────────────►◄
                └─-force─┘ └group_name─┘
```

Parameters

-force
(Optional) Specifies that the user group should be deleted even if there are users in the group.

Important: Using the force parameter might result in a loss of access. Use it only under the direction of your product support information.

group_id **|** *group_name*
(Required) The ID or name of the user group to be removed.

Description

Use the **rmusergrp** command to delete a user group.

You must have the Security Administrator role to create, delete, or change a user group.

User groups with users cannot normally be deleted. If you use the **force** parameter, the group is deleted and all of the users in that group are assigned to the Monitor group. Default user groups cannot be deleted, even if the **force** parameter is set.

An invocation example

```
rmusergrp support
```

The resulting output:

```
No feedback
```

testldapserver

Use the **testldapserver** command to test a Lightweight Directory Access Protocol (LDAP) server.

Syntax

```
►►── testldapserver ── ── -delim ── delimiter ─┬──────────────────────────────────────────────┬──────────►
                                                └─ -username ── user_name ─┬────────────────┬─┘
                                                                           └─┬─ -password ─┬┘
                                                                             └─ password ──┘

►─┬──────────────────────┬──────────────────────────────────────────────────────────────────────►◄
  ├─ ldap_server_id ─────┤
  └─  ldap_server_name ──┘
```

Parameters

-delim *delimiter*
(Optional) By default, in a concise view all columns of data are space-separated, with the width of each column set to the maximum possible width of each item of data. In a detailed view, each item of data is an individual row, and if displaying headers, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a one-byte character. Enter `-delim :` on the command line, and the colon character (**:**) separates all items of data in a concise view (for example, the spacing of columns does not occur); in a detailed view, the specified *delimiter* separates the data from its header

-username user_name
(Optional) Specifies the user name to test.

-password *password*
(Optional) Specifies the password to test. You can optionally specify the password with this

parameter. If you do not specify the password, the system prompts you for it before running the command and does not display the password that you type.

Note: The **-password** parameter is only valid if **-username** is specified. The actual password does not need to be supplied.

ldap_server_id|*ldap_server_name*
(Optional) Specifies the LDAP server ID or name to test.

Description

The **testldapserver** command allows three levels of testing:

- Server connection test (issue **testldapserver** without supplying username or password). This verifies that a connection can be established with the server while authenticating using the configured administrator credentials according to the LDAP configuration.
- Server connection, LDAP configuration, and user authorization test (issue **testldapserver** with a username). This verifies that:
  - A connection can be established with the server while authenticating using the configured administrator credentials.
  - The LDAP attributes are correctly configured on the system.
  - The user has been assigned a role.
- Server connection, LDAP configuration, and user authentication test (issue **testldapserver** with a username and password). This verifies that:
  - A connection can be established with the server while authenticating using the configured administrator credentials.
  - The user authenticates with the supplied password

No specific server errors indicates success.

Important: This command works whether or not LDAP authentication is selected or enabled with the **chauthservice** command.

Table 100 provides the attribute values that can be displayed as output view data.

Table 100. testldapserver attribute values

Attribute	Value
id	LDAP server ID
name	LDAP server name
error	Critical server error (or success, depending on situation) encountered

An invocation example with one LDAP server and no specific user information

```
testldapserver -delim ":" ldapserver1
```

The resulting output:

```
id:name:error
1:ldapserver1:CMMVC7075I The LDAP task completed successfully
```

An invocation example with all LDAP servers using a UPN

```
testldapserver -username bloggs@company.com -delim ":"
```

The resulting output:

```
id:name:error
0:ldapserver0:CMMVC6518E The task has failed because no roles
                         are defined for the current user on the system.
1:ldapserver1:CMMVC7075I The LDAP task completed successfully.
2:ldapserver2:CMMVC7075I The LDAP task completed successfully.
```

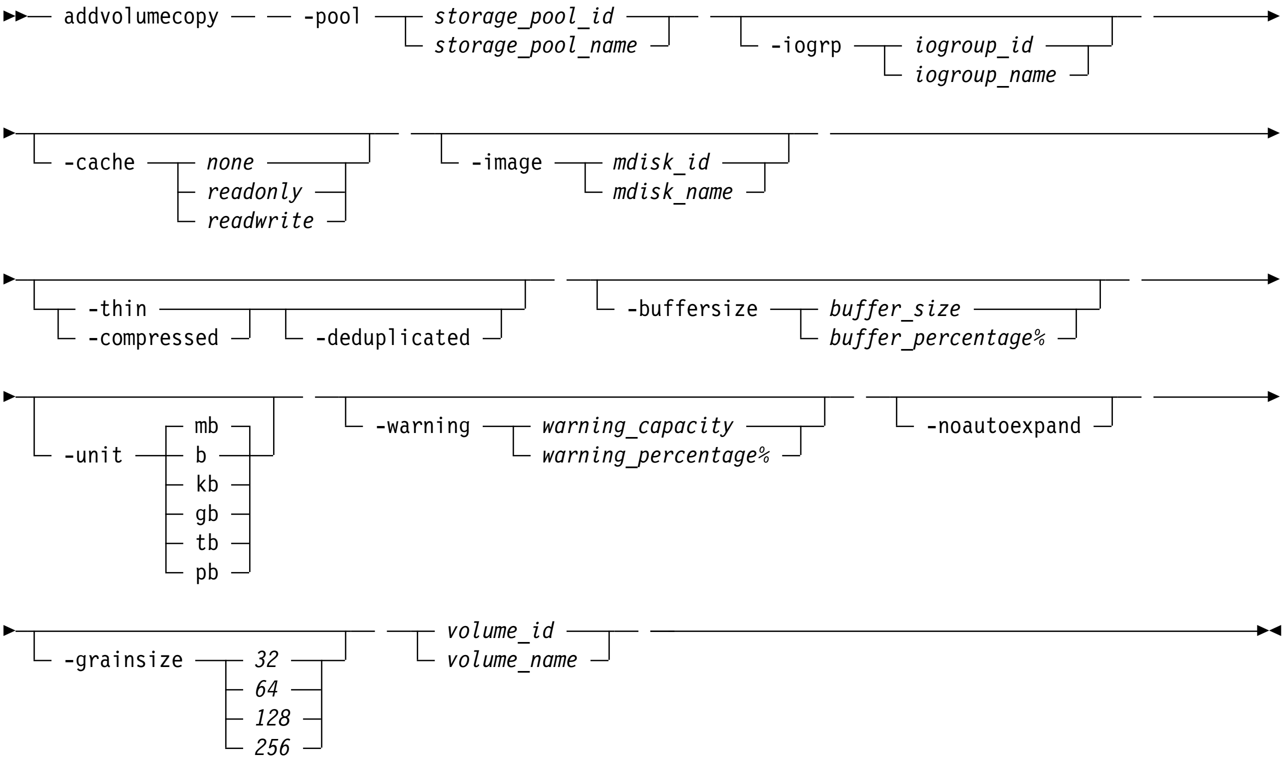
Chapter 27. Volume commands

Use the volume commands enable to work with volume options for the system.

addvolumecopy

Use the **addvolumecopy** command to add a copy to an existing volume. On a standard topology system, you can use this command to add a mirrored copy to an existing volume. On a stretched or HyperSwap topology system, you can use this command to convert an existing basic volume into a highly available volume by adding a copy of that volume at a second site.

Syntax



Parameters

-pool *storage_pool_id* | *storage_pool_name*
(Required) Specifies the storage pool in which to create the new volume copy.

Remember: For stretched and hyperswap topology systems, the site of storage pool must not be the same as the site of the existing volume copy.

-iogrp *iogroup_id* | *iogroup_name*
(Optional) Specifies the I/O group that the new volume copy is cached in.

Note: This parameter applies only when you are creating a HyperSwap volume and requires that the system topology be hyperswap.
The I/O group must be in the same site as the storage pool that is being specified.

-cache *none* | *readonly* | *readwrite*
(Optional) Specifies the caching options for the volume copy. Valid entries are:

- `readwrite` enables the cache for the volume.
- `readonly` disables write caching but allows read caching for a volume.
- `none` disables the cache mode for the volume.

Note: This parameter applies only when you are creating a HyperSwap volume and requires that the system topology be `hyperswap`.

-image *mdisk_id* | *mdisk_name*
(Optional) Specifies that the volume copy is to be created (on any topology) in image mode and it specifies which currently unused MDisk to use.

Note: For a `stretched` or `hyperswap` topology system, the MDisk site must match the storage pool site. If the storage pool is empty, the MDisk site must be `1` or `2`, and the MDisk site cannot be the same as the site of the existing volume copy.

-thin
(Optional) Specifies that the volume copy is to be created with thin provisioning. You cannot specify this parameter with **-compressed**.

Note: If you do not specify either **-thin** or **-compressed** parameters, the system creates a fully allocated volume copy.

-compressed
(Optional) Specifies that the volume copy is to be created compressed. You cannot specify this parameter with **-thin**.

-deduplicated
(Optional) Adds a deduplicated volume. If you specify **-deduplicated**, you must also specify **-thin** or **-compressed** because it applies only to thin or compressed volumes.

Note: Data deduplication works only with data reduction storage pools. You can only create deduplicated volumes and volume copies in an I/O group if no compressed volumes or volume copies exist in regular storage pools.

-buffersize *buffer_size* | *buffer_percentage%*
(Optional) Specifies the pool capacity the volume attempts to reserve as a buffer for thin-provisioned and compressed volumes. You must specify either **-thin** or **-compressed** with this parameter.

-warning *warning_capacity* | *warning_percentage%*
(Optional) Specifies a threshold at which a warning error log is generated for the volume copy. A warning is generated when the used disk capacity on the thin-provisioned or compressed copy exceeds the specified threshold. You can specify the threshold by using *warning_capacity* to specify a size, which defaults to MB unless the **-unit** parameter is specified.

Note: You can also specify a *warning_percentage%* to use a percentage of the volume size. If you do not specify a warning threshold, a default value of `80%` is used. To disable warnings, specify `0`.

You must specify either **-thin** or **-compressed** with this parameter.

-unit b | kb | mb | gb | tb | pb
(Optional) Specifies the data units for the **-buffersize** and **-warning** parameters.

-noautoexpand
(Optional) Specifies that the volume copy not automatically expand as it is written to; the available buffer capacity decreases as the used capacity increases. The copy goes offline if the buffer capacity is consumed.

The buffer capacity can be increased by specifying `expandvdisksize -rsize`. You must specify either **`-thin`** or **`-compressed`** with this parameter. If you do not specify this keyword, the copy automatically expands as it is written to.

`-grainsize` *`32`* **|** *`64`* **|** *`128`* **|** *`256`*
(Optional) Sets the grain size (KB) for a thin-provisioned volume. If you are using the thin-provisioned volume in a FlashCopy map, use the same grain size as the map grain size for best performance. If you are using the thin-provisioned volume directly with a host system, use a small grain size. The grain size value must be `32`, `64`, `128`, or `256` KB. The default is `256` KB.

`volume_id` **|** *`volume_name`*
(Required) Specifies the volume to add the volume copy to.

Description

Use the **`addvolumecopy`** command to add a copy to an existing volume. The new volume copy is synchronized with the current copy.

Note: A volume cannot have volume copies in different storage pools if cloud backup is enabled on the volume.

On some node types, you can create a compressed volume copy in a data reduction storage pool for an I/O group. A compressed volume copy in a data reduction pool can only be created in an I/O group with V5030, V7000, or SVC node types. You can create thin-provisioned volume copies on any node type. Volumes can also have fully allocated volume copies in data reduction storage pools.

You cannot specify **`-buffersize`** if the volume copy is created in a data reduction storage pool. Specify **`-thin`** or **`-compressed`** to enable thin provisioning or compression.

You cannot specify **`-noautoexpand`** when you create thin-provisioned or compressed volume copies from a data reduction storage pool.

You cannot create a volume copy that is a thin-provisioned or compressed volume in a data reduction storage pool, and the volume caching mode is `none` or `readonly`. You must specify **`chvdisk`** to change the volume caching mode to `readwrite`.

You cannot specify **`-warning`** for a thin-provisioned or compressed volume copy in a data reduction storage pool.

You cannot specify **`-grainsize`** for thin-provisioned and compressed volume copies in data reduction storage pools. This type of volume copy is created with a size of 8 KB.

Thin-provisioned or compressed volume copies in data reduction pools cannot be created if the data reduction storage pool is offline and requires recovery. If the recovery is still in progress, you must wait until the recovery is complete and the pool is in `online` state.

On a standard topology system, you can use this command to add a mirrored copy to an existing volume. On a stretched or HyperSwap topology system, you can use this command to convert an existing basic volume into a highly available volume by adding a copy of that volume to a second site.

A volume copy cannot be created in the same site as an existing copy of the volume. This command automatically adds the caching I/O group to the access I/O group set of the volume.

Scenario 1

If the I/O group contains:

- At least one 8 GB node.

- At least one thin-provisioned or compressed volume in a data reduction pool.
- And you try to set the FlashCopy bitmap size for that I/O group to at least 1.5 GB.

The command fails due to insufficient resources available.

Scenario 2

When a thin-provisioned or compressed volume is created within a data reduction pool, the pool must have enough capacity to create more volumes that track SCSI unmap operations from the host. If this capacity is not available, the command fails.

Scenario 3

Volumes cannot be created in a data reduction pool if offline thin-provisioned or compressed volumes exist in a data reduction pool, either because of thin provisioning (out of space or corruption), or a component underneath thin provisioning is holding a volume in the pool offline.

Add a volume copy to an existing volume

```
addvolumecopy -pool 2 volume5
```

The detailed resulting output:

```
No feedback
```

Add a thin-provisioned volume copy to an existing volume

```
addvolumecopy -pool site2pool1 -thin 0
```

The detailed resulting output:

```
No feedback
```

Add a fully allocated image-mode volume copy

```
addvolumecopy -image mdisk12 -pool 3 volume2
```

The detailed resulting output:

```
No feedback
```

Add a thin-provisioned volume copy

```
addvolumecopy -pool paulgilbert17 -thin thinvdisk3
```

The detailed resulting output:

```
No feedback
```

An invocation example to add a deduplicated volume copy

```
addvolumecopy -pool datareductionpool10 -thin -deduplicated deduplicatedvolume6
```

The resulting output:

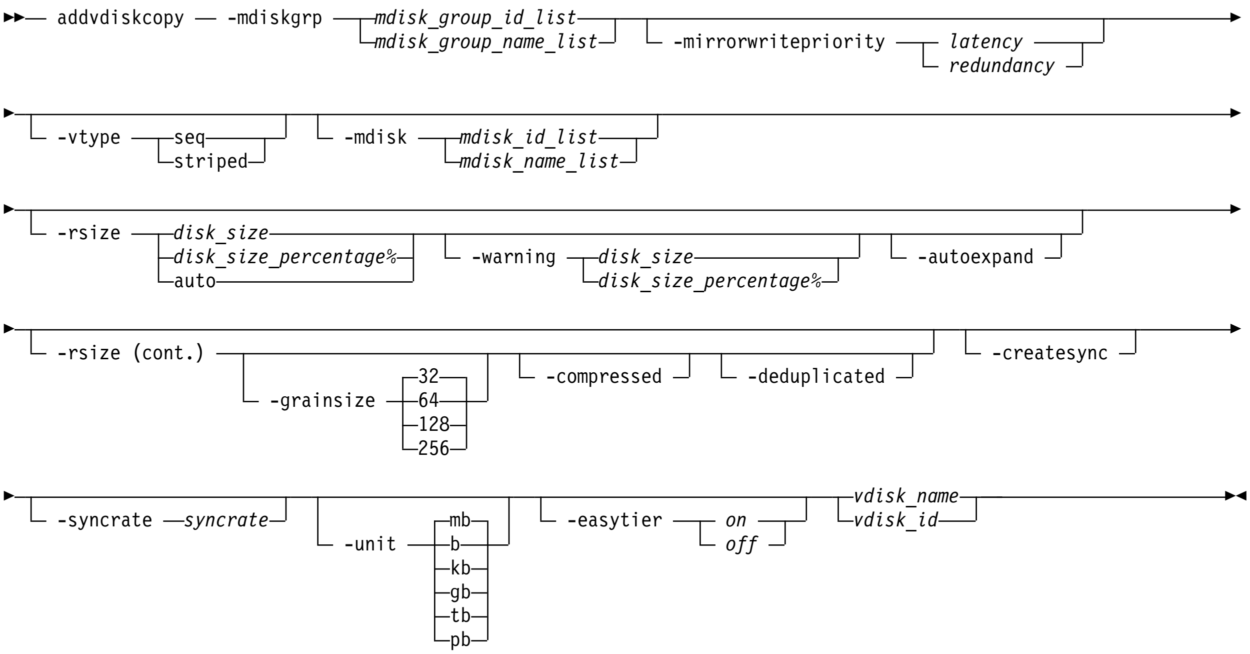
```
Vdisk [6] copy [1] successfully created
```

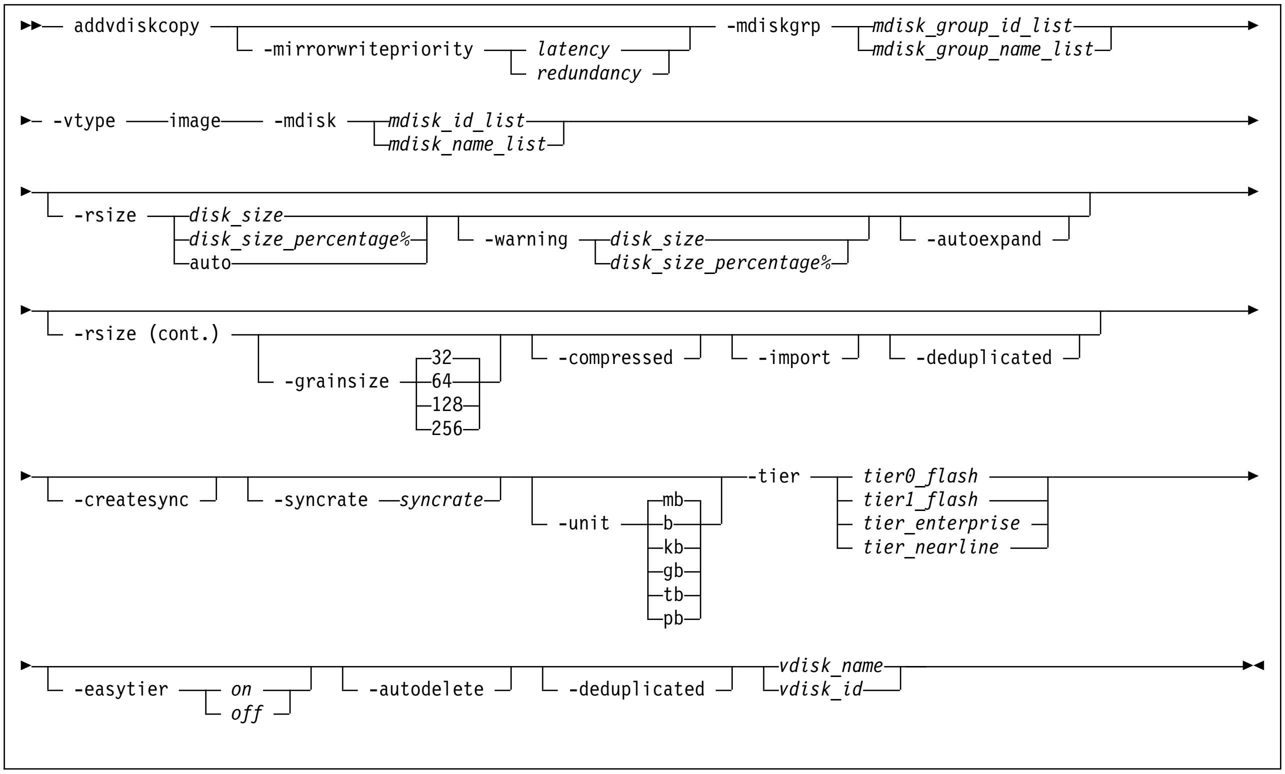
addvdiskcopy

Use the **addvdiskcopy** command to add a copy to an existing volume, which changes a nonmirrored volume into a mirrored volume. On a system with a `hyperswap` topology, use the **addvolumecopy** command to convert an existing volume to a HyperSwap volume by adding a copy at a second site.

Note: The first syntax diagram depicts the addition of a sequential or striped mode volume. The second syntax diagram depicts the addition of an image mode volume.

Syntax





Parameters

-mdiskgrp *mdisk_group_id_list* | *mdisk_group_name_list*
(Required) Specifies the storage pools to use to create copies for the volume. You must specify a group for each copy that is being added.

Note: If the MDisk group is from a child pool, **-vtype** must be `striped`.

-mirrorwritepriority *latency* | *redundancy*
(Optional) Specifies how to configure the mirror write algorithm priority.

1. Choosing *latency* means a copy that is slow to respond to a write input/output (I/O) becomes unsynchronized, and the write I/O completes if the other copy successfully writes the data.
2. Choosing *redundancy* means a copy that is slow to respond to a write I/O synchronizes completion of the write I/O with the completion of the slower I/O to maintain synchronization.

-vtype seq | striped | image
(Optional) Specifies the virtualization type for the copy: sequential, striped, or image. The type can be different than the virtualization types for other copies on the volume. The default virtualization type is **striped**. If you specify the **-rsize auto** option or the **-import** option, you must also specify the **-vtype image** option.

Note: You cannot create an image or sequential mode copy from a child pool or data reduction pools.

-mdisk *mdisk_id_list* | *mdisk_name_list*
(Optional) Specifies one or more managed disks (MDisks). For sequential and image mode copies, you must specify a single MDisk that has sufficient free extents. For image mode copies, the MDisk must be in unmanaged mode. For sequential mode copies the MDisk must be in the managed mode.

-syncrate *syncrate*
(Optional) Specifies the copy synchronization rate. A value of zero prevents synchronization. For the supported **-syncrate** values and their corresponding rates, see Table 102 on page 651.

If not specified, the current value is unchanged.

-createsync
(Optional) Suppresses the synchronization of the new volume copy with the primary copy. Using this parameter can cause data corruption if the primary copy fails and leaves an unsynchronized secondary copy to provide data. Using this parameter can cause loss of read stability in unwritten areas if the primary copy fails, data is read from the primary copy, and then different data is read from the secondary copy.

Note: You cannot specify **-createsync** for a volume that is fast formatting.

-rsize *disk_size* **|** *disk_size_percentage%* **| auto**
(Optional) Makes the copy thin-provisioned and specifies the real size of the copy. Specify the *disk_size* | *disk_size_percentage* value by using an integer, or an integer immediately followed by the percent character (%). The default units for *disk_size* are megabytes (MB). To specify different units, use the **-unit** parameter. The **auto** option creates a volume copy that uses the entire size of the MDisk; if you specify the **-rsize auto** option, you must also specify the **-vtype image** option.

-deduplicated
(Optional) Adds a deduplicated volume. If you specify **-deduplicated**, you must also specify **-rsize** because it applies only to thin-provisioned or compressed volumes.

Note: Data deduplication works only with data reduction storage pools. You can only create deduplicated volumes and volume copies in an I/O group if no compressed volumes or volume copies exist in regular storage pools.

-compressed
(Optional) Adds exactly one copy to an existing volume that already has (only) one copy a volume, and enables compression. Requires the **-rsize** parameter also be specified.

Remember:

- You cannot specify this parameter with the **-grainsize** parameter.
- When you specify this parameter with the **-import** parameter, you must specify **-rsize auto**.

-warning *disk_size* **|** *disk_size_percentage%*
(Optional) Requires that the **-rsize** parameter also be specified. Generates a warning when the used disk capacity on the thin-provisioned copy first exceeds the specified threshold. You can specify a *disk_size* integer, which defaults to megabytes (MB) unless the **-unit** parameter is specified; or you can specify a *disk_size*%, which is a percentage of the volume size. If **-autoexpand** is enabled, the default value for **-warning** is 80% of the volume capacity. If **-autoexpand** is not enabled, the default value for warning is 80% of the real capacity. To disable warnings, specify **0**.

-autoexpand
(Optional) Requires that the **-rsize** parameter also be specified. Specifies that thin-provisioned copies automatically expand their real capacities by allocating new extents from their storage pool. If the **-autoexpand** parameter is specified, the **-rsize** parameter specifies a capacity that is reserved by the copy. It protects the copy from going offline when its storage pool runs out of space by allowing it to consume this reserved space first.

-grainsize 32 | 64 | 128 | 256
(Optional) Requires that the **-rsize** parameter also be specified. Sets the grain size (KB) for a thin-provisioned volume copy. The grain size value must be 32, 64, 128, or 256 KB. The default is 256 KB.

-unit b | kb | mb | gb | tb | pb
(Optional) Specifies the data units for the **-rsize** and **-warning** parameters.

-import
(Optional) Imports an image mode disk that contains a thin-provisioned volume into the clustered system (system). Requires that the **-rsize** and **-vtype image** parameters also be specified.

-tier *tier0_flash* | *tier1_flash* | *tier_enterprise* | *tier_nearline*
(Optional) Specifies the MDisk tier when an image mode copy is added.

tier0_flash
Specifies a `tier0_flash` hard disk drive or an external MDisk for the newly discovered or external volume.

tier1_flash
Specifies a `tier1_flash` (or flash drive) hard disk drive or an external MDisk for the newly discovered or external volume.

tier_enterprise
Specifies a `tier_enterprise` hard disk drive or an external MDisk for the newly discovered or external volume.

tier_nearline
Specifies a `tier_nearline` hard disk drive or an external MDisk for the newly discovered or external volume.

-easytier *on* | *off*
(Optional) Determines whether the IBM Easy Tier function is allowed to move extents for this volume. If a volume copy is striped and not being migrated, see the settings in Table 101.

Table 101. Storage pool Easy Tier settings

Storage pool Easy Tier setting	Number of tiers in the storage pool	Volume copy Easy Tier setting	Volume copy Easy Tier status
Off	One	Off	inactive (see note 1 on page 649)
Off	One	On	inactive (see note 1 on page 649)
Off	Two	Off	inactive (see note 1 on page 649)
Off	Two	On	inactive (see note 1 on page 649)
Measure	One	Off	measured (see note 2 on page 649)
Measure	One	On	measured (see note 2 on page 649)
Measure	Two	Off	measured (see note 2 on page 649)
Measure	Two	On	measured (see note 2 on page 649)
Auto	One	Off	measured (see note 2 on page 649)
Auto	One	On	measured (see note 2 on page 649)
Auto	Two	Off	balanced (see note 3 on page 649)
Auto	Two	On	active (see note 4 on page 649)
On	One	Off	measured (see note 2 on page 649)
On	One	On	balanced (see note 3 on page 649)

Table 101. Storage pool Easy Tier settings (continued)

Storage pool Easy Tier setting	Number of tiers in the storage pool	Volume copy Easy Tier setting	Volume copy Easy Tier status
On	Two	Off	measured (see note 2)
On	Two	On	active (see note 4)
Notes: 1. When the volume copy status is `inactive`, no IBM Easy Tier functions are enabled for that volume copy. 2. When the volume copy status is `measured`, the IBM Easy Tier function collects usage statistics for the volume but automatic data placement is not active. 3. When the volume copy status is `balanced`, the IBM Easy Tier function enables performance-based pool balancing for that volume copy. 4. When the volume copy status is `active`, the IBM Easy Tier function operates in automatic data placement mode for that volume.			

If the volume copy is in image or sequential mode or is being migrated, the volume copy IBM Easy Tier status is `measured` instead of `active`.

The default Easy Tier setting for a storage pool is `auto`, and the default Easy Tier setting for a volume copy is `on`. If the setting is `on`, it means that Easy Tier functions except pool performance balancing are disabled for storage pools with a single tier, and that automatic data placement mode is enabled for all striped volume copies in a storage pool with two or more tiers.

-autodelete
(Optional) Specifies the primary copy is deleted after the secondary copy is synchronized.

j*vdisk_name* | *vdisk_id*
(Required) Specifies the volume to add the volume copy to, either by ID or by name.

Description

The **addvdiskcopy** command adds a copy to an existing volume, which changes a nonmirrored volume into a mirrored volume. Use the **mkdiskgrp** parameter to specify the storage pools that provide storage for the copy; the **lsmdiskgrp** command lists the available storage pools and the amount of available storage in each group.

The **addvdiskcopy** command can be specified with a file system volume, but must be used with the same storage pool for that volume.

Remember: Only compressed copies are allowed to be added to file system volumes.
The **addvdiskcopy** command adds a different volume copy, such as a copy created from an uncompressed to compressed conversion or a compressed to uncompressed conversion.

Note: A volume cannot have volume copies in different storage pools if cloud snapshot is enabled on the volume.

A thin-provisioned or compressed volume copy in a data reduction storage pool must not be a sequential or image mode volume. On some node types, you can create a compressed volume copy in a data reduction storage pool for an I/O group. A compressed volume copy in a data reduction pool can only be created in an I/O group with V5030, V7000, or SVC node types. You can create thin-provisioned volume copies on any node type. Use the **-autoexpand**parameter to create thin-provisioned or compressed volume copies from a data reduction storage pool. Volumes can also have fully allocated volume copies in data reduction storage pools.

You cannot create a volume copy that is a thin-provisioned or compressed volume in a data reduction storage pool, and the volume caching mode is `none` or `readonly`. You must specify **chvdisk** to change the volume caching mode to `readwrite`.

You cannot specify **-warning** for a thin-provisioned or compressed volume copy in a data reduction storage pool.

For thin-provisioned and compressed volume copies in data reduction storage pools, the Easy Tier mode for the volume is taken from the data reduction storage pool. The Easy Tier mode cannot be configured on these volume types.

You cannot specify **-grainsize** for thin-provisioned and compressed volume copies in data reduction storage pools. This type of volume copy is created with a size of 8 KB.

Thin-provisioned or compressed volume copies in data reduction pools cannot be created if the data reduction storage pool is offline and requires recovery. If the recovery is still in progress, you must wait until the recovery is complete and the pool is in `online` state.

An encryption key cannot be used when you add an image mode MDisk. To use encryption (when the MDisk has an encryption key), the MDisk must be self-encrypting.

Remember: You cannot add a volume copy if the volume to be copied is being formatted.

The virtualization types are defined as follows:

sequential (seq)
This policy requires the **-mdisk** parameter with a single managed disk as its argument. This MDisk must be in the managed mode.

It creates the volume by using extents from the given managed disk (assuming enough free extents exist on the managed disk).

striped
The `striped` policy is the default policy. If the **-vtype** parameter is not specified, this policy is used in its default form. That is, all managed disks in the storage pool are used to create the volume. The striping is at an extent level; one extent from each managed disk in the group is used. For example, a storage pool with 10 managed disks uses one extent from each managed disk, then it uses the 11th extent from the first managed disk, and so on.

If the **-mdisk** parameter is also specified, you can supply a list of managed disks to use as the stripe set. This list can include two or more managed disks from the same storage pool. The same circular algorithm is used across the striped set. However, a single managed disk can be specified more than once in the list. For example, if you enter `-m 0:1:2:1`, the extents are from the following managed disks: 0, 1, 2, 1, 0, 1, 2, and so forth. All MDisks that are specified in the **-mdisk** parameter must be in managed mode.

image
This policy allows image mode volumes to be created when a managed disk already has data on it, perhaps from a previrtualized subsystem. When an image mode volume is created, it directly corresponds to the (previously unmanaged) managed disk that it was created from; therefore, volume logical block address (LBA) *x* equals managed disk LBA *i*. You can use this command to bring a nonvirtualized disk under the control of the system. After it is under the control of the system, you can migrate the volume from the single managed disk. When it is migrated, the volume is no longer an image mode volume.

You can add image mode volumes to an already populated storage pool with other types of volumes, such as a striped or sequential.

Note: An image mode copy must be at least as large as the volume that it is being added to, but any capacity beyond the size of the volume is not accessible.

The command returns the ID of the newly created volume copy.

Create the first compressed volume copy for an I/O group to activate compression. You cannot create or move a compressed volume copy to an I/O group that contains (at least) one node that does not support compressed volumes. You must use another I/O group, but note that it does not affect moving to the recovery I/O group.

Important:

- If the volume (or volume copy) is a target of a FlashCopy mapping with a source volume in an `active-active` relationship, the new storage pool must be in the same site as the source volume.
- If this command is used for a volume that is a master volume, an auxiliary volume, or a change volume of an `active-active` relationship, the new copy must be created in a storage pool of the same site as the existing volume copy.
- When you add an image mode copy, the site information for the MDisk being added must be well-defined and match the site information for any other MDisks in the storage pool.

The rate at which the volume copies resynchronize after loss of synchronization can be specified by using the **`-syncrate`** parameter. Table 102 provides the relationship of the *syncrate* value to the data copied per second.

Note: These settings also affect the initial rate of formatting.

Table 102. Relationship between the syncrate value and the data copied per second

User-specified *syncrate* attribute value	**Data copied/sec**
1 - 10	128 KB
11 - 20	256 KB
21 - 30	512 KB
31 - 40	1 MB
41 - 50	2 MB
51 - 60	4 MB
61 - 70	8 MB
71 - 80	16 MB
81 - 90	32 MB
91 - 100	64 MB

Scenario 1

If the I/O group contains:

- At least one 8 GB node.
- At least one thin-provisioned or compressed volume in a data reduction pool.
- And you try to set the FlashCopy bitmap size for that I/O group to at least 1.5 GB.

The command fails due to insufficient resources available.

Scenario 2

When a thin-provisioned or compressed volume is created within a data reduction pool, the pool must have enough capacity to create more volumes that track SCSI unmap operations from the host. If this capacity is not available, the command fails.

Scenario 3

Volumes cannot be created in a data reduction pool if offline thin-provisioned or compressed volumes exist in a data reduction pool, either because of thin provisioning (out of space or corruption), or a component underneath thin provisioning is holding a volume in the pool offline.

An invocation example

```
addvdiskcopy -mdiskgrp 0 -easytier off vdisk8
```

The resulting output:

```
Vdisk [8] copy [1] successfully created
```

An invocation example for specifying storage pools

```
addvdiskcopy -mdiskgrp 0 -vtype image -mdisk 13 -tier tier0_flash -easytier off vdisk9
```

The resulting output:

```
Vdisk [9] copy [1] successfully created
```

An invocation example for configuring a mirror write algorithm priority

```
addvdiskcopy -mdiskgrp 0 -mirrorwritepriority latency vdisk9
```

The resulting output:

```
Vdisk [9] copy [1] successfully created
```

An invocation example for adding a compressed volume copy

```
addvdiskcopy -mdiskgrp 1 -rsize 10% -compressed vdisk2
```

The resulting output:

```
Vdisk [2] copy [1] successfully created
```

An invocation example for adding a compressed volume copy

```
addvdiskcopy -mdiskgrp 0 -vtype image -mdisk 13 -tier tier_nearline vdisk9
```

The resulting output:

```
Vdisk [9] copy [1] successfully created
```

An invocation example to add a deduplicated volume copy

```
addvdiskcopy -mdiskgrp datareductionpool10 -rsize 0 -autoexpand -deduplicated deduplicatedvolume6
```

The resulting output:

```
Vdisk [6] copy [1] successfully created
```

addvdiskaccess

Use the **addvdiskaccess** command to add an I/O group (or groups) to the set of I/O groups in which a volume can be made accessible to hosts.

Syntax

```
►►— addvdiskaccess — —  -iogrp ─┬─ iogrp_id_list ──┬─ ─┬─ vdisk_id ──┬──────────────────►◄
                                └─ iogrp_name_list ┘   └─ vdisk_name ┘
```

Parameters

-iogrp *iogrp_id_list* | *iogrp_name_list*
(Required) Specifies a list of I/O groups to add to the I/O group volume access set.

vdisk_id | *vdisk_name*
(Required) Specifies the volume to which to add access through the specified I/O groups.

Description

If an I/O group is already a member of the access set, no error is generated and no action is taken for that I/O group. All host mappings for the volume are added to the I/O groups in the list. The **-force** option is not required to extend additional mappings to other I/O groups.

When an I/O group is added to the access set, it creates access to the volume from the hosts that are mapped to the volume from the nodes in the I/O group. If the volume is mapped twice, it is also mapped twice through all additional I/O groups.

You can add I/O groups to the volume access list if they are mapped to iSCSI hosts. This means that iSCSI hosts can access volumes that are accessible through multiple I/O groups (as well as a single I/O group).

Remember: The **-addvdiskaccess** command fails if:

- Any host (for which the volume has a host mapping) is not associated with an I/O group in the list
- The host volume mapping limit is exceeded
- The amount of extra mappings added exceeds the clustered system limit for host volume mappings

Two mappings are created if a host is mapped to a volume with two I/O groups. Hosts are limited to 512 host-to-volume mappings, which means a host can be mapped to:

- 512 volumes in a single I/O group
- 256 volumes across two I/O groups
- 64 volumes across four I/O groups

The command fails if any host mapped to the volume is detected as a host system that does not support volumes mapped from multiple I/O groups.

An invocation example

This example adds I/O group 2 to the volume access set for DB_Volume:

```
addvdiskaccess -iogrp 2 DB_Volume
```

The resulting output:

```
No feedback
```

An invocation example

This example adds I/O groups 2 and 3 to the volume access set for volume ID 3:

```
addvdiskaccess -iogrp 2:3 3
```

The resulting output:

```
 No feedback
```

analyzevdisk

Use the **analyzevdisk** command to queue or cancel volume analysis.

Syntax

```
►►── analyzevdisk ── ─┬─────────┬─ ─┬─ vdisk_id ───┬──────────────────────────────────────►◄
                      └─ -cancel┘   └─ vdisk_name ─┘
```

Parameters

-cancel
(Optional) Cancels an ongoing compression estimation.

vdisk_id | *vdisk_name*
(Required) Specifies the volume ID or name to queue for analysis.

Description

This command queues or cancels volume analysis. The order is based on the *vdisk_id* value.

Important: You cannot specify `analyzevdisk -cancel` for a volume that is not currently being analyzed (or is queued for analysis).

You can schedule an offline volume for analysis (no error message is displayed). The volume remains scheduled until it is back online and is analyzed according to its *vdisk id* value.

A concise invocation example to queue `vdisk 0` for analysis

```
analyzevdisk 0
```

The detailed resulting output:

```
No feedback
```

A concise invocation example to dequeue or cancel an ongoing analysis for `vdisk 0`

```
analyzevdisk -cancel 0
```

The detailed resulting output:

```
No feedback
```

analyzevdiskbysystem

Use the **analyzevdiskbysystem** command to schedule all existing volumes in system for free space analysis.

Syntax

```
►►── analyzevdiskbysystem ── ─┬─────────┬─ ──────────────────────────────────────────────►◄
                              └─ -cancel┘
```

Parameters

-cancel
(Optional) Cancels a scheduled or pending compression estimate.

Description

This command schedules all existing volumes in system for free space analysis.

Volumes that are created after you specify the command are not evaluated. Use **analyzevdisk** to evaluate specific volumes.

A concise invocation example to queue `vdisk 0` for analysis

```
analyzevdiskbysystem
```

The detailed resulting output:

```
No feedback
```

A concise invocation example to dequeue or cancel an ongoing analysis for `vdisk 0`

```
analyzevdiskbysystem -cancel
```

The detailed resulting output:

```
No feedback
```

backupvolume

Use the **backupvolume** command to create a volume snapshot.

Syntax

```
►►─ backupvolume ─ ┬─────────┬─ ─┬─ volume_name ─┬──────────────────────────────►◄
                   └─ -full ─┘   └─ volume_id ───┘
```

Parameters

-full
(Optional) Specifies that the snapshot generation for the volume should be a full snapshot.

volume_name | *volume_id*
(Required) Specifies the volume name or ID for the volume being backed up. The value for the volume name must be an alphanumeric string and the value for the volume ID must be a number.

Description

This command creates a volume snapshot.

The command completes when the volume snapshot is taken, and the snapshot is asynchronously transferred to the cloud system.

Note: If a volume belongs to a volume group, you must specify **backupvolumegroup** instead of **backupvolume**.

An invocation example for creating a full snapshot generation that has existing volume snapshots on the cloud system

```
backupvolume -full vdisk7
```

The resulting output:

```
No feedback
```

An invocation example for creating a backup of a volume for the first time

```
backupvolume neymar7
```

The resulting output:

```
No feedback
```

An invocation example for creating a backup of a volume that has existing snapshots in the cloud

```
backupvolume jvardy6
```

The resulting output:

```
No feedback
```

An invocation example for creating a full snapshot for a volume that has existing snapshots in the cloud

```
backupvolume -full lmessi1
```

The resulting output:

```
No feedback
```

backupvolumegroup

Use the **backupvolumegroup** command to create a new snapshot for all of the volumes in a volume group.

Syntax

```
►►— backupvolumegroup — ┬─────────┬─ ┬─ volumegroup_name ─┬─ ──────────────────────────────►◄
                        └─ -full ─┘  └─ volumegroup_id ───┘
```

Parameters

-full
(Optional) Specifies a full backup for volume group members.

volumegroup_name | *volumegroup_id*
(Required) Specifies a volume group ID or name for the volume to back up. The value must be a number for the volume group ID and an alphanumeric string for the volume group name.

Description

This command creates a new snapshot for all of the volumes in a volume group.

The command completes as soon as a snapshot of the volume group is taken. The backup is asynchronously transferred to the cloud. If any of the volume members have a backup or restore in progress, any new volume group backup cannot be taken. The volume backup on each volume member needs to be enabled by using the **chvdisk** command to enable the volume backup for entire volume group.

An invocation example

To create a backup of a volume group for the first time:

```
backupvolumegroup volgroup1
```

The resulting output:

```
No feedback
```

An invocation example

To create a backup of a volume group that has existing backups in the cloud:

```
backupvolumegroup volgroup1
```

The resulting output:

```
No feedback
```

An invocation example

To create a full backup generation for a volume group that has existing backups in the cloud:

```
backupvolumegroup -full volgroup1
```

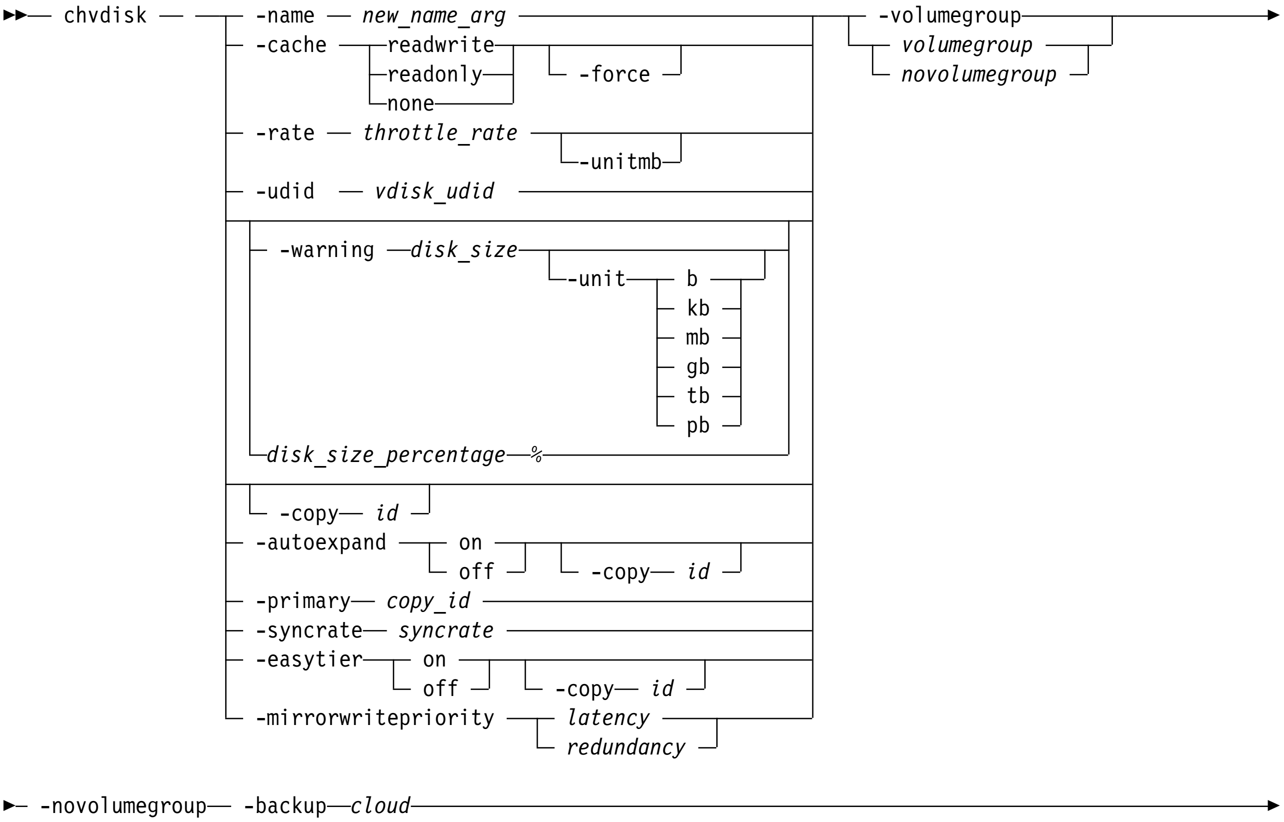
The resulting output:

```
No feedback
```

chvdisk

Use the **chvdisk** command to modify the properties of a volume, such as the disk name, I/O governing rate, or unit number. You can also change IBM Easy Tier settings.

Syntax



```
►─┬──────────────────────────────────────────────────────────────────────┬─┬─ vdisk_name ─┬──────►◄
  ├─ -enable── -account──┬─cloud_account_name─┬──┬──────────────────────────────┬─┤ └─ vdisk_id ───┘
  │                      └─cloud_account_id───┘  └─ -backupgrainsize──┬─64──┬──┘ │
  │                                                                   └─256─┘     │
  └─ -disable─────────────────────────────────────────────────────────────────────┘
```

Parameters

-name *new_name_arg*
(Optional) Specifies a new name to assign to the volume. You cannot use this parameter with the **-rate** or **-udid** parameters. This parameter is required if you do not use the **-rate** or **-udid** parameters.

Note: Do not use this parameter with file system volumes.

-cache readwrite | readonly | none
(Optional) Specifies the caching options for the volume. Valid entries are:

- Use *readwrite* to enable the cache for the volume.
- Use *readonly* to disable write caching and allow read caching for a volume.
- Use *none* to disable the cache mode for the volume.

The default is *readwrite*.

-force
(Optional) The **force** parameter can be used only for changing the caching mode. Use the **force** parameter with the **cache** parameter to specify that you want the system to change the cache mode of the volume even if the I/O group is offline. This option overrides the cache flush mechanism.

Attention: If the **force** parameter is used for changing the caching mode, the contents of the cache are discarded and the volume might be corrupted by the loss of the cached data. This corruption might occur if the system is able to destage all write data from the cache or not. Use the **force** parameter with caution.

Important: Using the force parameter might result in a loss of access. Use it only under the direction of your product support information.

-rate *throttle_rate* **-unitmb**
(Optional) Specifies the I/O governing rate for the volume, which caps the amount of I/O that is accepted. The default *throttle_rate* units are I/Os. By default the *throttle_rate* is disabled. To change the *throttle_rate* units to megabits per second (MBps), specify the **-unitmb** parameter. The governing rate for a volume can be specified by I/Os or by MBps, but not both. However, you can set the rate to I/Os for some volumes and to MBps for others. When the input/output operations per second (IOPS) limit is configured on a volume, and it is smaller than 100 IOPS, the throttling logic rounds it to 100 IOPS. Even if throttle is set to a value smaller than 100 IOPs, the actual throttling occurs at 100 IOPs.

Note: To disable the throttling on a specific volume, set the `throttle_rate` value to zero.

You cannot use this parameter with the **-name** or **-udid** parameters.

-udid *vdisk_udid*
(Optional) Specifies the unit number (**-udid**) for the disk. The *vdisk_udid* is an identifier that is required to support OpenVMS hosts; no other systems use this parameter. Valid options are a decimal number in the range 0 - 32,767, or a hexadecimal number from 0 to 0x7FFF. A hexadecimal number must be preceded by `0x` (for example, `0x1234`). If you do not use the **-udid** parameter, the default **-udid** is `0`.

You cannot use this parameter with the **-name** parameters.

-warning *disk_size* **|** *disk_size_percentage%*

(Optional) Generates a warning when the used disk capacity on the thin-provisioned copy first exceeds the specified threshold. You can specify a *disk_size* integer, which defaults to MBs unless the **-unit** parameter is specified; or you can specify a *disk_size%*, which is a percentage of the volume size. To disable warnings, specify `0` or `0%`.

-unit b | kb | mb | gb | tb | pb

(Optional) Specifies the data units to use for the **-warning***disk_size* parameter. The default unit value is MB.

-autoexpand on | off

(Optional) Specifies whether thin-provisioned volume copies automatically expand their real capacities by allocating new extents from their storage pool. To use this parameter, the volume must be thin-provisioned.

-copy *id*

(Optional) Specifies the copy to apply the changes to. You must specify this parameter with the **-autoexpand** or **-warning** parameter. The **-copy** parameter is required if the specified volume is mirrored and only one volume copy is thin-provisioned. If both copies are thin-provisioned and the **-copy** parameter is not specified, the specified **-autoexpand** or **-warning** parameter is set on both copies.

-primary *copy_id*

(Optional) Specifies the primary copy. Changing the primary copy takes effect only when the new primary copy is online and synchronized. If the new primary is online and synchronized when the command is issued, the change takes effect immediately. You cannot change the volume's primary copy if that primary copy has its `autodelete` flag is set to `yes` (on).

Important: You cannot use this parameter with a volume that is fast formatting.

-syncrate *syncrate*

(Optional) Specifies the copy synchronization rate. A value of zero (`0`) prevents synchronization. The default value is `50`. See Table 103 on page 661 for the supported **-syncrate** values and their corresponding rates. Use this parameter to alter the rate at which the fully allocated volume or mirrored volume format before synchronization.

-easytier on | off

(Optional) Enables or disables the IBM Easy Tier function.

Attention: Thin-provisioned and compressed volumes in a data reduction pool always have IBM Easy Tier on, regardless of the pool setting.

-mirrorwritepriority *latency* **|** *redundancy*

(Optional) Specifies how to configure the mirror write algorithm priority. A change to the mirror write priority is reflected in the volume's view immediately and in the volume's behavior after all prior input and output (I/O) completes.

1. Choosing *latency* means a copy that is slow to respond to a write I/O becomes unsynchronized, and the write I/O completes if the other copy successfully writes the data.
2. Choosing *redundancy* means a copy that is slow to respond to a write I/O synchronizes completion of the write I/O with the completion of the slower I/O to maintain synchronization.

-volumegroup *volumegroup_name* **|** *volumegroup_id*

(Optional)

Specifies a new volume group for a volume. This parameter is mutually exclusive with **-novolumegroup**.

-novolumegroup

(Optional) Specifies that a volume does not belong in any volume group. This parameter is mutually exclusive with **-volumegroup**.

-backup *cloud*
(Optional) Specifies the cloud snapshot type to enable or disable. The value must be `cloud`.

-enable
(Optional) Enables the backup or snapshot type that is specified with the **-backup** parameter.

-disable
(Optional) Disables the backup or snapshot type that is specified with the **-backup** parameter.

-account *cloud_account_id* | *cloud_account_name*
(Optional) Specifies the cloud account to use for the volume. You must specify **-enable** with this parameter.

-backupgrainsize *64* | *256*
(Optional) Specifies the grain size (in KB) for volume mappings. The values are `64` and `256`. You must specify **-enable** to use this parameter.

You can enable a volume for a cloud snapshot with one account. You cannot enable cloud backup on a volume for a second time on the same or different cloud account.

You cannot turn off the cloud snapshot function if a snapshot in progress. Any snapshot that is in progress must complete or be canceled.

vdisk_name | *vdisk_id*
(Required) Specifies the volume to modify, either by ID or by name.

Description

The **chvdisk** command modifies a single property of a volume. To change the volume name and modify the synchronization rate, for example, you must issue the command twice. If the volume is offline, use **recovervdisk** command to recover the volume and bring it back online.

Important: To change the caching I/O group for a volume or preferred node, use the **movevdisk** command.

A thin-provisioned or compressed copy that is in a data reduction storage pool must enable **-autoexpand**. If a volume contains a copy that is in a data reduction storage pool, the cache mode must be set to `readwrite`.

A thin-provisioned or compressed copy that is in a data reduction storage pool cannot have a warning threshold set. To change the warning threshold, you must specify the **-copy**.

You can specify a new name or label. Then, you can use the new name to refer to the volume.

You can set a limit on the number of I/O transactions that is accepted for this volume. It is set in terms of I/Os per second or MBs per second. By default, no I/O governing rate is set when a volume is created.

Attention: All capacities, including changes, must be in multiples of 512 bytes. An error occurs if you specify a capacity that is not a multiple of 512, which can happen only when byte units are used. The default capacity is in MB.

When the volume is created, no throttling is applied to it. Use the **-rate** parameter to change it. To change the volume back to an unthrottled state, specify 0 (zero) with the **-rate** parameter.

For thin-provisioned and compressed volume copies in data reduction pools, the Easy Tier status is derived from the data reduction pool as the data is managed by a central data disk. Therefore, the Easy Tier mode cannot be turned off on these volume types. The thin-provisioned and compressed volumes in a data reduction pool always have Easy Tier on, regardless of the pool setting. The Easy Tier setting is a combination of pool and volume setting, as detailed in a table in the help for **mkvdisk**. Leaving easy tier always on for the volume allows the pool setting to be the one that switches it on or off. For fully allocated volumes in a data reduction pool, Easy Tier can be turned both on and off.

The rate at which the volume copies resynchronize after loss of synchronization can be specified by using the **-syncrate** parameter. This table provides the relationship of the *syncrate* value to the data copied per second.

Note: These settings also affect the initial rate of formatting.

Table 103. Relationship between the syncrate value and the data copied per second

User-specified *syncrate* attribute value	**Data copied/sec**
1 - 10	128 KB
11 - 20	256 KB
21 - 30	512 KB
31 - 40	1 MB
41 - 50	2 MB
51 - 60	4 MB
61 - 70	8 MB
71 - 80	16 MB
81 - 90	32 MB
91 - 100	64 MB

An invocation example

```
chvdisk -rate 2040 1
```

The following output is displayed:

```
No feedback
```

An invocation example

```
chvdisk -cache readonly 1
```

The following output is displayed:

```
No feedback
```

An invocation example

```
chvdisk -volumegroup 1 vdisk2
```

The following output is displayed:

```
No feedback
```

An invocation example

To enable a cloud snapshot for a volume, enter the following command:

```
chvdisk -backup cloud -enable -account myVardyj vdisk7
```

The following output is displayed:

```
No feedback
```

An invocation example

To disable a cloud snapshot for a volume, enter the following command:

```
chvdisk -backup cloud -disable vdisk7
```

The following output is displayed:

```
No feedback
```

chvolumegroup

Use the **chvolumegroup** command to change volume group properties.

Syntax

```
►►── chvolumegroup ── ┬───────────────────────────────┬─ ─┬─ volumegroup_name ─┬─ ──────────────►◄
                      └─ -name ── volumegroup_name ───┘   └─ volumegroup_id ───┘
```

Parameters

-name *volumegroup_name*
: (Optional) Specifies a new volume group name. The value must be an alphanumeric string.

volumegroup_name | *volumegroup_id*
: (Required) Specifies the volume group name or group ID for the volume you want to modify. The value must be a number for the volume group ID, and an alphanumeric string for the volume group name.

Description

This command changes volume group properties.

An invocation example

```
chvolumegroup -name newname1 1
```

The resulting output:

```
No feedback
```

expandvdisksize

Use the **expandvdisksize** command to expand the size of a volume by a given capacity.

Syntax

```
►►── expandvdisksize ── ┬─ -size  disk_size ──────────────────┬─ ─────────────────────────────►
                        └─ -rsize  disk_size─┬────────────────┤
                                             └─ -copy── id ───┘

►─┬──────────────────────────────────────┬─ ─┬──────────────┬─ ─┬─────────────────────┬─ ──────►
  └─ -mdisk ─┬─ mdisk_id_list ───────┬───┘   └─ -fmtdisk ───┘   └─ -unit ─┬─ b ──┬───┘
             └─ mdisk_name_list ─────┘                                    ├─ kb ─┤
                                                                          ├─ mb ─┤
                                                                          ├─ gb ─┤
                                                                          ├─ tb ─┤
                                                                          └─ pb ─┘
```

```
►─┬─ vdisk_name ─┬──────────────────────────────────────────────────────────►◄
  └─ vdisk_id ───┘
```

Parameters

-size *disk_size*
: (Required) Specifies the capacity by which the volume is expanded. Disk size is used with the value of the unit. All capacities, including changes must be in multiples of 512 bytes. An error occurs if you specify a capacity that is not a multiple of 512, which can only occur when byte units (**-unit b**) are used. However, an entire extent is reserved even if it is only partially used. The default *disk_size* unit is megabytes (MB). You cannot specify the **-size** parameter with the **-rsize** parameter. You must specify either **-size** or **-rsize**. If the volume is thin-provisioned, MDisks cannot be specified.

-rsize *disk_size*
: (Optional) Specifies the capacity by which to increase the real size of a thin-provisioned volume. Specify the *disk_size* value using an integer. Specify the unit for a *disk_size* integer using the **-unit** parameter; the default unit is megabytes (MB). The **-rsize** value can be greater than, equal to, or less than the size of the volume. You cannot specify the **-rsize** parameter with the **-size** parameter. You must specify either **-size** or **-rsize**.

-copy *id*
: (Optional) Specifies the copy to change the real capacity for. You must also specify the **-rsize** parameter; you can only modify the real capacity of a volume copy. The **-copy** parameter is required if the specified volume is mirrored and only one copy is thin-provisioned. If the volume is mirrored, both copies are thin-provisioned and **-copy** is not specified, both copies are modified by the same amount.

-mdisk *mdisk_id_list* | *mdisk_name_list*
: (Optional) Specifies the list of one or more MDisks to be used as the stripe set. The extents that expand the volume come from the specified list of MDisks. All MDisks in the list must be part of the same storage pool. The **-mdisk** parameter cannot be used if the specified volume is mirrored.

-fmtdisk
: (Optional) Specifies that the volume be formatted before use. This parameter formats the new extents that have been added to the volume as a result of the **expandvdisksize** command. The **expandvdisksize** command completes asynchronously if you use this parameter.

-unit b | kb | mb | gb | tb | pb
: (Optional) Specifies the *disk_size* unit for the **-size** or **-rsize** parameter. The default value is megabytes (MB).

vdisk_name | *vdisk_id*
: (Required) Specifies the volume to modify, either by ID or by name.

Description

Use the **expandvdisksize** command to expand the physical capacity that is allocated to a particular volume by the specified amount.

The command can also be used to expand the virtual capacity of a thin-provisioned volume without altering the physical capacity that is assigned to the volume. To change the capacity of a non-thin-provisioned volume, or the virtual capacity of a thin-provisioned volume, use the **-size** parameter. To change the real capacity of a thin-provisioned volume, use the **-rsize** parameter.

Note: For relationships that cannot be resized, you cannot expand the capacity of any volume in a Global Mirror or Metro Mirror relationship, regardless of whether it is a primary, secondary, or a Change Volume. To expand the capacity of a volume in a Global Mirror or Metro Mirror relationship:

1. Delete the relationship.

2. Increase the size of all the volumes. All volumes in a relationship must have the exact same size (virtual capacity).
3. Re-create the relationship with the larger volumes.

When the mirror is restarted, it will do a complete initial synchronization, replicating the entire primary volume to the secondary volume.

Note: You can expand the capacity of any volume in a Global Mirror or Metro Mirror relationship that is in `consistent_synchronized` state if those volumes are using thin-provisioned or compressed copies. You cannot expand the capacity for these types of volumes:

- Volumes in HyperSwap relationships or in Global Mirror relationships that are operating in `cycling` mode
- Volumes in relationships where a change volume configured
- Volumes that have a fully allocated volume copy

You can not expand the capacity of any volume in a FlashCopy mapping, regardless of whether it is a source or target, or what state the mapping is in. You can expand the capacity of a volume in a FlashCopy mapping:

1. Delete all the mappings in that FlashCopy tree. (There is a root source volume and some targets either directly or cascaded off of other targets - the entire tree must be deleted.)
2. Increase the size of all volumes in the original FlashCopy tree. All volumes in a tree must be the same size (virtual capacity).
3. Re-create all the FlashCopy mappings with the new larger volumes.

When a FlashCopy is restarted after being deleted (including if it is an incremental FlashCopy) the entire volume becomes part of any background copy because it is the start of a new mapping.

Note: The default capacity units are MB.

When a volume is expanded, the virtualization policy can change. Its mode becomes striped even if it was previously sequential. See the **mkvdisk** command for details of the virtualization policies.

To run the **expandvdisksize** command on a mirrored volume, all copies of the volume must be synchronized. The command formats all copies of a mirrored volume automatically.

Remember:

1. You cannot resize (expand) an image mode volume.
2. You cannot resize (expand) a volume that is part of a file system.
3. You cannot resize (expand) volume if that volume is being fast formatted. (Additionally, you cannot specify **shrinkvdisksize** to resize (shrink) for a volume that is fast formatting.)
4. You can not resize (expand) a volume if cloud snapshot is enabled on that volume.
5. You cannot specify `expandvdisksize -rsize` to expand (resize) a thin or compressed volume copy that is in a data reduction pool.
6. You cannot specify `expandvdisksize -mdisk` to resize (expand) a volume when a volume is being migrated.

You must expand both volumes in a relationship to maintain full operation of the system. To perform this:

1. Expand the secondary volume by the required additional capacity
2. Expand the primary volume by the required additional capacity

An invocation example to increase the capacity of `vdisk1` by `2048` bytes using extents from two MDisks (and to format the new part of the volume)

```
expandvdisksize -size 2048 -unit b -mdisk mdisk0:mdisk1 -fmtdisk vdisk1
```

The resulting output:

```
No feedback
```

An invocation example to increase the capacity of `vdisk1` by `100` MB using extents from two MDisks (and to format the new part of the volume)

```
expandvdisksize -size 100 -unit mb -mdisk mdisk0:mdisk1 -fmtdisk vdisk1
```

The resulting output:

```
No feedback
```

An invocation example to increase the real capacity of thin-provisioned `vdisk2` by `100` MB without changing the virtual capacity (and to spread the extents across all MDisks in the storage pool)

```
expandvdisksize -rsize 100 -unit mb vdisk2
```

The resulting output:

```
No feedback
```

An invocation example to increase the real capacity of thin-provisioned volume copy id `1` of mirrored volume `vdisk3` by `100` MB

```
expandvdisksize -rsize 100 -unit mb -copy 1 vdisk3
```

The resulting output:

```
No feedback
```

lsdependentvdisks

Use the **`lsdependentvdisks`** command to view which volumes go offline if you remove a specific piece of hardware from the system.

Syntax

```
►►── lsdependentvdisks ──┬──────────┬──┬──────────────────────┬───────────────────►
                         └─ -nohdr ─┘  └─ -delim ──delimiter──┘

►──┬─ -node ──┬─node_name─┬───────────────────────────┬───────────────────────►◄
   │          └─node_id───┘                           │
   ├─ -controller ──┬─controller_name_list─┬──────────┤
   │                └─controller_id_list───┘          │
   ├─ -mdisk ──┬─mdisk_name_list─┬────────────────────┤
   │           └─mdisk_id_list───┘                    │
   ├─ -drive ──drive_id_list──────────────────────────┤
   └─ -enclosure ──enclosure_id──┬────────────────────────┬┘
                                 └─ -canister ──canister_id─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter **-delim :** on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-node *node_name* | *node_id*
(Optional) Specifies the node for which volume dependency is required.

-controller *controller_name_list* | *controller_id_list*
(Optional) Specifies the controllers for which volume dependency is required.

-mdisk *mkdisk_name_list* | *mkdisk_id_list*
(Optional) Specifies the MDisks for which volume dependency is required.

-drive
(Optional) Specifies the drives for which volume dependency is required. There is a maximum of 128 entries.

-enclosure *enclosure_id*
(Optional) Specifies the enclosure for which volume dependency is required. You can remove a control enclosure without affecting your other data.

-canister *canister_id*
(Optional) Specifies an enclosure canister. The possible values are 1 and 2.

Description

Use this command to view which volumes go offline if you remove a specific piece of hardware from the system. Use this command to determine which volumes are affected before undergoing maintenance.

An invocation example

```
lsdependentvdisks -delim : -drive 0:1
```

The resulting output:

```
vdisk_id:vdisk_name
4:vdisk4
5:vdisk5
```

Note: This means that if drives 0 and 1 are removed, then volume vdisk4 and volume vdisk5 go offline.

lshostvdiskmap

Use the **lshostvdiskmap** command to display a list of volumes that are mapped to a host. These volumes are the volumes that are recognized by the specified host.

Syntax

```
►►── lshostvdiskmap ── ┬──────────┬── ┬──────────────────────┬── ┬─────────────┬──────────────►◄
                       └─ -nohdr ─┘   └─ -delim ── delimiter ─┘   ├─ host_id ───┤
                                                                  └─ host_name ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

host_id | *host_name*
(Optional) Specifies the host in terms of its ID or name. The command displays a list of all the volumes that are mapped to the specified host and the Small Computer System Interface (SCSI) ID by which they are mapped. If you do not specify a host ID or name , the command displays a list of all recognized volume mappings.

Description

This command displays a list of volume IDs and names. These volumes are the volumes that are mapped to the specified host; that is, they are visible to the specified host. The SCSI LUN ID is also displayed. This SCSI LUN ID is the ID by which the volume is recognized by the host.

Each volume that is exported by the clustered system (system) is assigned a unique virtual path (VPATH) number. This number identifies the volume and determines which volume corresponds to the volume that the hosts recognize. This procedure must be completed by using the command-line interface.

For a specific volume based on which operating system and multipath software are used, you can use different commands to determine the VPATH serial number. For example, issuing **datapath query device** finds the VPATH serial number for volumes that are mapped to `AIX sddpcm`.

Find the host that is defined to the system that corresponds with the host that you are working with.

1. The worldwide port names (WWPNs) are an attribute of the host bus adapter (HBA). You can find the WWPNs by looking at the device definitions that are stored by your operating system. For example, on AIX® they are in the Object Data Manager (ODM), in Windows® they are in the Device Manager details for the specified HBA.
2. Verify which host is defined to the system that these ports belong to. The ports are stored as part of the detailed view, so you must list each host in turn by issuing the following command:

   ```
   lshost host_name | host_id
   ```

   where *host_name* | *host_id* is the name or ID of the host. Check for matching WWPNs.

   Note: Consider this when you name your hosts. For example, if the actual host is called `orange`, also name the host that is defined to the system `orange`.

When you define the *hostname* and the *vpath serial number* to the system, issue the following command:

```
lshostvdiskmap hostname
```

where *hostname* is the name of the host. A list is displayed. Look for the volume UID that matches the *vpath serial number* and record the volume name or ID.

The command returns the following values:

id Indicates the host ID in the output for **lshostvdiskmap**.

name Indicates the host name in the output for **lshostvdiskmap**.

SCSI_id
Specifies the SCSI ID.

host_cluster_id
Indicates the unique ID for a host system.

host_cluster_name
Indicates the unique name for a host system.

vdisk_id
Indicates the ID of the volume.

vdisk_name
Indicates the name of the volume.

vdisk_UID
Indicates the UID of the volume.

IO_group_id
Indicates the ID of the input/output (I/O) group in which the host volume mapping exists.

IO_group_name
Specifies the name of I/O group in which the host volume mapping exists.

An invocation example

```
lshostvdiskmap -delim : 2
```

The resulting output:

```
id:name:SCSI_id:host_id:host_name:vdisk_id:vdisk_name:vdisk_UID:IO_group_id:IO_group_name
2:host2:0:5:vardy1:10:vdisk10:6005076801958001500000000000000A:0:iogrp0
2:host2:1:4:vardy2:11:vdisk11:6005076801958001500000000000000B:1:iogrp1
2:host2:2:3:vardy3:12:vdisk12:6005076801958001500000000000000C:0:iogrp0
2:host2:3:2:vardy4:13:vdisk13:6005076801958001500000000000000D:1:iogrp1
2:host2:4:1:vardy5:14:vdisk14:6005076801958001500000000000000E:1:iogrp0
```

An invocation example

```
lshostvdiskmap 0
```

The resulting output:

```
id name   SCSI_id host_id host_name vdisk_UID                        IO_group_id IO_group_name mapping_type host_cluster_id
0  vdisk0 0       1       hvlab02c2 6005076801D901A3F8000000000000 0           io_grp0       shared       0
0  vdisk0 0       4       vmlab02c1 6005076801D901A3F8000000000000 0           io_grp0       shared       0
0  vdisk0 0       5       vmlab02c2 6005076801D901A3F8000000000000 0           io_grp0       shared       0
0  vdisk0 0       24      vmlab14c1 6005076801D901A3F8000000000000 0           io_grp0       shared       0
0  vdisk0 0       25      vmlab14c2 6005076801D901A3F8000000000000 0           io_grp0       shared       0
0  vdisk0 0       26      vmlab15   6005076801D901A3F8000000000000 0           io_grp0       private
```

lsmetadatavdisk

Use the **lsmetadatavdisk** command to display the information for metadata volume.

Syntax

```
►►── lsmetadatavdisk ──┬─────────┬──┬────────────────────┬──────────────►◄
                       └─ -nohdr ─┘  └─ -delim ── delimiter ─┘
```

Parameters

-nohdr

(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*

(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

Description

This command displays the information for metadata volume.

Table 104 provides the attribute values that can be displayed as output view data.

Table 104. lsmetadatavdisk output

Attribute	Description
vdisk_id	Indicates the ID of the metadata volume.
vdisk_name	Indicates the name of the metadata volume.
status	Indicates the running status of the metadata volume.

An invocation example

To display information for a metadata volume:

```
lsmetadatavdisk
```

The resulting output:

```
vdisk_id          2
vdisk_name        vdisk2
status            online
```

lsrepairsevdiskcopyprogress

The **lsrepairsevdiskcopyprogress** command lists the repair progress for thin-provisioned volume copies or compressed volume copies.

Syntax

```
►►──lsrepairsevdiskcopyprogress── ┬─────────┬─ ┬────────────────────┬─┬───────────┬──►
                                  └─ -nohdr ─┘ └─ -delim ── delimiter ─┘ └─ -copy ── id ─┘
```

```
►─┬─────────────┬──────────────────────────────────────────────────────────►◄
  ├─vdisk_name─┤
  └─vdisk_id───┘
```

Parameters

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-copy *id*
(Optional) Lists the repair progress for the specified copy.

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

vdisk_name | *vdisk_id*
(Optional) Specifies the volume name or ID to list repair progress for. You must specify this parameter last on the command line. If you do not enter this parameter, the command lists progress for all thin-provisioned copies in the clustered system.

Description

The **lsrepairsevdiskcopyprogress** command lists the repair progress for thin-provisioned or compressed copies for the specified volume. If you do not specify a volume, the command lists the repair progress for all thin-provisioned or compressed copies in the clustered system.

Remember: Run this command after you specify the **repairsevdiskcopy** command, which you must run as required by the fix procedures or by your product support information.

The command returns values for the following volume copy attributes:

task Specifies the active task.

- `repairing` indicates repair of a thin-provisioned volume copy
- `compressed_repairing` indicates repair of a compressed volume copy.

progress
Specifies the task completion percentage.

estimated_completion_time
Specifies the expected duration of the task in the format `YYMMDDHHMMSS` (or blank if the estimated completion is unknown).

An invocation example

```
lsrepairsevdiskcopyprogress –delim :
```

The resulting output:

```
id:name:copy id:task:progress:estimated_completion_time
0:vdisk0:0:repairing:50:070301120000
0:vdisk0:1:repairing:51:070301120000
1:vdisk1:0:repairing:32:070301153500
```

An invocation example

```
lsrepairsevdiskcopyprogress –delim : vdisk0
```

The resulting output:

```
id:name:copy id:task:progress:estimated_completion_time
0:vdisk0:0:repairing:50:070301120000
0:vdisk0:1:repairing:51:070301120000
```

An invocation example

```
lsrepairsevdiskcopyprogress –delim : -copy 1 vdisk0
```

The resulting output:

```
id:name:copy id:task:progress:estimated_completion_time
0:vdisk0:1:repairing:51:070301120000
```

lsrepairvdiskcopyprogress

Use the **lsrepairvdiskcopyprogress** command to display the progress of volume repairs and validations.

Syntax

```
►►— lsrepairvdiskcopyprogress — ───────────── ─────────────────────── ────────────────►
                                └ -nohdr ┘    └ -delim — delimiter ┘  └ -copy — id ┘

►─┬─────────────┬────────────────────────────────────────────────────────────────►◄
  ├─vdisk_name─┤
  └─vdisk_id───┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim : on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-copy *id*
(Optional) Specifies the volume copy ID to list repair progress for. If you do not specify this parameter, progress is displayed for all copies.

vdisk_name | *vdisk_id*
(Optional) Specifies the volume name or ID to list repair progress for. You must specify this parameter last on the command line.

Description

The **lsrepairvdiskcopyprogress** command displays the progress of repairs and validations that are made to mirrored volumes. Use this command to track progress after you run the **repairvdiskcopy** command. You can specify a volume copy by using the **-copy** parameter. To display the volumes that have two or more copies with an active task, specify the command with no parameters; it is not possible to have only one volume copy with an active task.

The command displays progress for the following types of volume copies:

- All volume copies display the same task; validate, medium, or resync, depending on the specified parameter.
- All volume copies display the same percentage and estimated completion time.
- If specified, non-mirrored volumes are displayed as a single copy with a blank task; they are not displayed in the full concise view.
- Once a task completes, the task is blank for all copies.
- If the task is blank, the percentage and the completion time are also blank.

The command returns values for the following volume repair attributes:

vdisk_id
Indicates the volume ID.

vdisk_name
Indicates the volume name.

copy_id
Indicates the system-assigned identifier for the volume copy.

task
Indicates the active task. The values can be `repairing` or `compressed_repairing`.

progress
Indicates the task completion percentage. This value is 0 when task is in `compressed_repairing` state.

estimated_completion_time
Indicates the expected time (duration) the task completion time. The value is in the `YYMMDDHHMMSS` format, and is blank if the duration is not known.

An invocation example

```
lsrepairvdiskcopyprogress –delim :
```

The resulting output:

```
vdisk_id:vdisk_name:copy id:task:progress:estimated_completion_time
0:vdisk0:0:medium:50:070301120000
0:vdisk0:1:medium:50:070301120000
```

An invocation example

```
lsrepairvdiskcopyprogress –delim : vdisk0
```

The resulting output:

```
vdisk_id:vdisk_name:copy id:task:progress:estimated_completion_time
0:vdisk0:0:medium:50:070301120000
0:vdisk0:1:medium:50:070301120000
```

An invocation example

```
lsrepairvdiskcopyprogress –delim : -copy 0 vdisk0
```

The resulting output:

```
vdisk_id:vdisk_name:copy id:task:progress:estimated_completion_time
0:vdisk0:0:medium:50:070301120000
```

An invocation example showing a compressed volume copy and a TP volume copy being repaired

```
lsrepairvdiskcopyprogress
```

The resulting output:

```
vdisk_id vdisk_name copy_id task                progress estimated_completion_time
0        vdisk0     0       repairing           50       070301120000
2        vdisk2     1       compressed_repairing 0       070301080102
```

lssevdiskcopy

Use the **lssevdiskcopy** command to list the thin-provisioned copies of the specified volumes.

Syntax

```
►►──lssevdiskcopy── ─┬────────┬─ ─┬────────┬─ ─┬────────────────────┬─ ──────────────►
                     └─ -nohdr ─┘   └─ -bytes ─┘   └─ -delim ── delimiter ─┘

►─┬─────────────┬─ ─┬────────────────┬─┬─vdisk_name─┬──────────────────────────────────►◄
  └─ -copy ── id ─┘   └─ -filtervalue? ─┘ └─vdisk_id───┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If no data is available to be displayed, headings are not displayed.

-bytes
(Optional) Displays all capacities as bytes. Capacity values that are displayed in units other than bytes might be rounded.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim : on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-copy *id*
(Optional) Specifies the volume copy to list thin-provisioned copies for. You must specify a *vdisk_name* | *vdisk_id* value with this parameter.

-filtervalue?
(Optional) Displays a list of valid filter attributes. The following filters for the **lssevdiskcopy** command are valid:

- mdisk_grp_id
- mdisk_grp_name
- overallocation

- autoexpand
- grainsize
- 1 deduplicated_copy

vdisk_name | *vdisk_id*
(Optional) Specifies the volume name or ID to list thin-provisioned copies for. You must specify this parameter last on the command line. If you do not enter this parameter, the command lists all thin-provisioned copies in the clustered system.

Description

The **lssevdiskcopy** command lists all thin-provisioned copies of the specified volume. If you do not specify a volume, the command lists all thin-provisioned volume copies in the clustered system.

The command provides a concise view of the thin-provisioned properties of the selected volume copies. Run the **`lsvdiskcopy`** command to see a concise view of the properties that are common to thin-provisioned and non-thin-provisioned volume copies. See the description of the **`lsvdisk`** command for a description of the fields that are shown in the view.

The command returns values for the following volume copy attributes:

copy_id
Indicates a system-assigned identifier for the volume copy. The value can be `0` or `1`.

status
Indicates the system status. The value can be `online` or `offline`. A copy is offline if all nodes cannot access the storage pool that contains the copy.

sync
Indicates whether the volume copy is synchronized.

auto_delete
Indicates that the primary copy is deleted after the secondary copy is synchronized. The values are `yes` or `no`.

primary
Indicates whether the volume copy is the primary copy. A volume has exactly one primary copy. The value can be `yes` or `no`.

mdiskgrp_id
Indicates the ID of the storage pool that the volume copy belongs to.

mdiskgrp_name
Indicates the name of the storage pool that the volume copy belongs to.

type
Indicates the virtualization type of the volume. The value can be `striped`, `sequential`, or `image`.

mdisk_id
Indicates the MDisk ID that is used for sequential and image mode volumes.

mdisk_name
Indicates the MDisk name that is used for sequential and image mode volumes.

fast_write_state
Indicates the cache state of the volume copy. The value can be `empty`, `not_empty`, `corrupt`, or `repairing`. The value is always empty for non-thin-provisioned copies. A cache state of `corrupt` indicates that the volume is thin-provisioned and requires repair that is initiated by a **`recovervdisk`** command or the **`repairsevdiskcopy`** command.

used_capacity
Indicates the portion of `real_capacity` that is being used to store data. For non-thin-provisioned copies, this value is the same as the volume capacity. If the volume copy is thin-provisioned, the value increases from zero to the `real_capacity` value as more of the volume is written to.

1 **Remember:** This field is blank for thin-provisioned and compressed volume copies in a data
1 reduction pool.

real_capacity
Indicates the amount of physical storage that is allocated from a storage pool to this volume copy. If the volume copy is not thin-provisioned, the value is the same as the volume capacity. If the volume copy is thin-provisioned, the value can be different.

1 **Remember:** This field is blank for thin-provisioned and compressed volume copies in a data
1 reduction pool.

free_capacity
Indicates the difference between the `real_capacity` and `used_capacity` values.

1 **Remember:** This field is blank for thin-provisioned and compressed volume copies in a data
1 reduction pool.

overallocation
Expressed as a percentage, indicates the ratio of volume capacity to `real_capacity` values. This value is always `100` for non-thin-provisioned volumes.

1 **Remember:** This field is zero for thin-provisioned and compressed volume copies in a data
1 reduction pool.

autoexpand
Indicates whether `autoexpand` is enabled on a thin-provisioned volume. The value can be `on` or `off`.

warning
Expressed as a percentage, for thin-provisioned volume copies only, indicates that a warning is generated when the ratio of `used_capacity` to volume capacity reaches the specified level.

1 **Remember:** This field is zero for thin-provisioned and compressed volume copies in a data
1 reduction pool.

grainsize
For thin-provisioned volume copies, indicates the grain size that is chosen for the volume copy when it was created.

1 **Remember:** This field is blank for compressed volume copies in regular storage pools.

se_copy
Indicates whether the copy is thin-provisioned. The value can be `yes` or `no`.

easy_tier
Indicates whether Easy Tier is permitted to manage the pool.

Note:

1. If easy_tier is `on`, then easy_tier_status can take on any value.
2. If easy_tier is `off`, then easy_tier_status is `measured` or `inactive`.

easy_tier_status
Indicates which Easy Tier functions are active for the volume copy:

- `active` indicates that a pool is being managed by Easy Tier to provide tier management performance-based pool balancing. For example, extents of this volume copy can be moved for performance (automatic data placement).
- `inactive` indicates that no Easy Tier function is active.
- `measured` indicates that statistics are being gathered for this volume copy, but no extents are moved.

- balanced indicates that a pool is being managed by Easy Tier to provide performance-based pool balancing (for example, extents can be moved).

Storage pool Easy Tier setting	Number of tiers in the storage pool	Volume copy Easy Tier setting	Volume copy Easy Tier status
Off	One	Off	inactive (see note 2)
Off	One	On	inactive (see note 2)
Off	Two	Off	inactive (see note 2)
Off	Two	On	inactive (see note 2)
Measure	One	Off	measured (see note 3)
Measure	One	On	measured (see note 3)
Measure	Two	Off	measured (see note 3)
Measure	Two	On	measured (see note 3)
Auto	One	Off	measured (see note 3)
Auto	One	On	balanced (see note 4)
Auto	Two	Off	active (see note 5)
Auto	Two	On	measured (see note 3)
On	One	Off	measured (see note 3)
On	One	On	balanced (see note 4)
On	Two	Off	measured (see note 3)
On	Two	On	active (see note 5)

Note:

1. If the volume copy is in image or sequential mode or is being migrated, the volume copy Easy Tier(tm) status is measured instead of active.
2. When the volume copy status is inactive, no Easy Tier(tm) functions are enabled for that volume copy.
3. When the volume copy status is measured, the Easy Tier(tm) function collects usage statistics for the volume but automatic data placement is not active.
4. When the volume copy status is balanced, the Easy Tier(tm) function enables performance-based pool balancing for that volume copy.
5. When the volume copy status is active, the Easy Tier(tm) function operates in automatic data placement mode for that volume.
6. The default Easy Tier(tm) setting for a storage pool is auto, and the default Easy Tier(tm) setting for a volume copy is on. These settings mean that Easy Tier(tm) functions except pool performance balancing are disabled for storage pools with a single tier, and that automatic data placement mode is enabled for all striped volume copies in a storage pool with two or more tiers.

tier
Indicates which tier information is being reported:

- tier0_flash
- tier1_flash
- tier_enterprise
- tier_nearline

tier_capacity
Indicates the total MDisk capacity that is assigned to the volume in the tier.

Note: For thin-provisioned copies, the capacity by tier is the real capacity.

compressed_copy
Indicates whether the volume copy is compressed.

uncompressed_used_capacity
1 For compressed volume copies, indicates the amount of data written to the volume copy before
1 compression. This field is blank for volume copies in a data reduction storage pool.

used_capacity_before_reduction
1 Indicates the total amount of data written to a thin-provisioned or compressed volume copy in a
1 data reduction storage pool before data reduction occurs. This field is blank for fully allocated
1 volume copies and volume copies not in a data reduction pool.

parent_mdisk_grp_id
Indicates the physical storage pool ID that the volume extents are allocated from. This value is a numeric string (in the range 0 - 127 characters) or blank.

parent_mdisk_grp_name
Indicates the physical storage pool name that the volume extents are allocated from. This value is an alphanumeric string (in the range 1 - 63 characters) or blank.

encrypt
Indicates whether the volume and its copies are encrypted. The values are `yes` or `no`.

1 **deduplicated_copy_count**
1 Indicates the number of data deduplicated volume copies.

1 **deduplicated_copy**
1 Indicates whether the volume copy is data deduplicated. The values are:
1 - `yes`
1 - `no`

An invocation example

```
lssevdiskcopy –delim :
```

The following output is displayed:

```
vdisk_id:vdisk_name:copy_id:mdisk_grp_id:mdisk_grp_name:capacity:used_capacity:real
_capacity:
free_capacity:overallocation:autoexpand:warning:grainsize:se_copy:compressed_copy
:uncompressed_used_capacity
0:vv1:0:0:ppp:16.00GB:2.00GB:2.01GB:6.00GB:796:off:20:32:no:yes:3.27GB
1:se1:0:0:ppp:16.00GB:1.00GB:4.00GB:15.00GB:400:off:20:32:yes:no:1.0GB:yes:no:1.0GB
1:se1:1:0:ppp:16.00GB:2.00GB:2.01GB:14.00GB:796:off:45:256:no:yes:4.46GB
```

An invocation example

```
lssevdiskcopy -delim : -copy 0 0
```

The following output is displayed:

```
vdisk_id:0
vdisk_name:vv1
capacity:16.00GB
copy_id:0
status:online
sync:yes
auto_delete:yes

primary:yes
mdisk_grp:1
mdisk_grp name:mdisk_group_1
type:striped
mdisk_id:
mdisk_name:
fast_write_state:not_empty
used_capacity:2.00GB
real_capacity:2.01GB
free_capacity:6.00GB
```

```
overallocation:796
autoexpand:on
warning:25
grainsize:256
se_copy:yes
easy_tier:on
easy_tier_status:active

tier:tier0_flash
tier_capacity:1.63TB
tier:tier1_flash
tier_capacity:1.63TB
tier:tier_enterprise
tier_capacity:0.00MB
tier:tier_nearline
tier_capacity:0.00MB
tier_capacity:64.00MB
tier:ssd
tier_capacity:2.00GB
compressed_copy:yes
uncompressed_used_capacity:3.27GB
used_capacity_before_reduction

parent_mdisk_grp_id:10
parent_mdisk_grp_name:pool10
encrypt:yes
```

An invocation example

```
lssevdiskcopy -copy 0 –delim : vv1
```

The following output is displayed:

```
vdisk_id:0
vdisk_name:vv1
capacity:16.00GB
copy_id:0
status:online
sync:yes
auto_delete:yes

primary:yes
mdisk_grp_id:1
mdisk_grp_name:mdisk_group_1
type:striped
mdisk_id:
mdisk_name:
fast_write_state:empty
used_capacity:2.00GB
real_capacity:8.00GB
free_capacity:6.00GB
overallocation:200
autoexpand:on
warning:25
grainsize:256
se_copy:yes
easy_tier:off
easy_tier_status:inactive

tier:tier0_flash
tier_capacity:1.63TB
tier:tier1_flash
tier_capacity:1.63TB
tier:tier_enterprise
tier_capacity:0.00MB
tier:tier_nearline
tier_capacity:0.00MB
compressed_copy:no
```

```
uncompressed_used_capcaity:8.00GB
parent_mdisk_grp_id:10
parent_mdisk_grp_name:pool10
encrypt:yes
used_capacity_before_reduction
```

1 An invocation example

```
1 lsvdisk 0
```

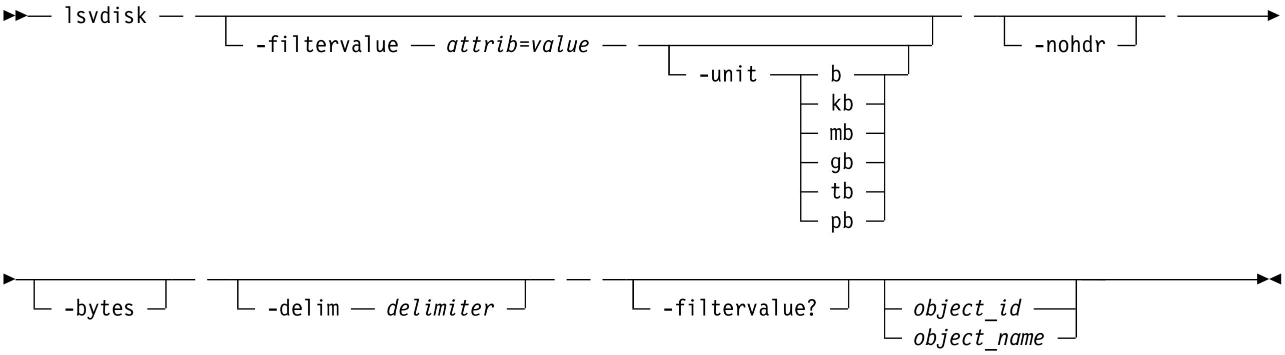
1 The following output is displayed:

```
1 vdisk_id:0
1 vdisk_name:vv1
1 ...
1 deduplicated_copy_count:1
1 ..
1 ...
1 copy_id 0
1 ...
1 deduplicated_copy:yes
1 used_capacity_before_reduction:12.54GB
```

lsvdisk

Use the **lsvdisk** command to display a concise list or a detailed view of volumes that are recognized by the system.

Syntax



Parameters

-filtervalue *attrib=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed. If a capacity is specified, the units must also be included.

-unit b | kb | mb | gb | tb | pb
(Optional) Specifies the data units for the **-filtervalue** parameter.

Note: **-unit** must be used with **-filtervalue**.

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-bytes

(Optional) Displays all capacities as bytes. Capacity values that are displayed in units other than bytes might be rounded. When you filter a capacity value, use a unit of bytes, **-unit b**, for exact filtering. For thin-provisioned copies, the capacity by tier contains the real capacities.

-delim *delimiter*

(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-filtervalue?

Displays a list of valid filter attributes. The following filters for the **lsvdisk** command are valid:

- access_IO_group_count
- backup_status
- capacity
- cloud_backup_enabled
- cloud_account_id
- cloud_account_name
- compressed_copy_count
- copy_count
- deduplicated_copy_count 1
- fast_write_state
- FC_id
- fc_map_count
- FC_name
- filesystem
- function
- id
- IO_group_id
- IO_group_name
- mdisk_grp_name
- mdisk_grp_id
- mirror_write_priority
- name
- owner_type
- owner_id
- owner_name
- preferred_node_id
- RC_change
- RC_id
- RC_name
- restore_status
- se_copy_count
- status
- type

- vdisk_UID
- volume_group_id
- volume_group_name
- volume_id
- volume_name

Note: It is not possible to filter the **lsvdisk** command with mdisk_grp_name=many to identify mirrored volumes. Instead, filter on copy_count=2.

object_id | *object_name*
(Optional) Specifies the name or ID of an object. When you use this parameter, the detailed view of the specific object is returned and any value that is specified by the **-filtervalue** parameter is ignored. If you do not specify the *object_id* | *object_name* parameter, the concise view of all objects that match the filtering requirements that are specified by the **-filtervalue** parameter are displayed.

Description

This command displays a concise list or a detailed view of attributes for all volumes and volume copies in the system.

The volume is offline and unavailable if one of the following actions takes place:

- Both nodes in the I/O group are missing.
- None of the nodes in the I/O group that are present can access the volume.
- All synchronized copies for this volume are in storage pools that are offline.
- The volume is formatting.

If you have a degraded volume and all of the associated nodes and MDisks are online, refer to your product support information for assistance. A volume is reported as degraded if any of the following occurs:

- One of the nodes in the I/O group is missing.
- One of the nodes in the I/O group cannot access all the MDisks in the storage pool that the volume spans. In this case, MDisks are shown as degraded and the fix procedures for MDisks must be followed to resolve the problem.
- The fast write cache pins data for one or more volumes in the I/O group and is unable to perform a failback until the situation is resolved. An error log indicating that the cache contains pinned data is displayed. Follow the fix procedures for this event log to resolve the problem. The following situations are the most common causes of pinned data:
  - One or more volumes in an I/O group is offline due to an asymmetric failure and contains pinned data in the cache. Asymmetric failures can occur because of fabric faults or misconfiguration, back-end controller faults or misconfiguration, or because repeated errors cause the system to exclude access to a MDisk through one or more nodes.
  - One or more volumes in an I/O group is offline due to a problem with a FlashCopy mapping.
  - A thin-provisioned disk runs out of space.

You can encrypt volumes and volume copies. The volume is encrypted if all volume copies are also encrypted.

Note: This means that during a migration of a volume (with one copy) between an encrypted and unencrypted storage pool the value is no.
The volume is not encrypted if the storage pool has a value of encrypt:no, including if the volume's extents are part of an encrypted MDisk or sequential volume.

The command returns values for the following volume attributes:

IO_groups_id
Indicates the I/O group (ID) that the volume belongs to.

IO_groups_name
Indicates the I/O group (name) that the volume belongs to.

status Indicates the status. The value can be online, offline, or degraded.

For online HyperSwap volumes the scope of offline copy is included with the status information. For offline volumes the auxiliary copy is included with the status information even if the copy associated with that is online.

mdisk_grp_id
Indicates the ID of the storage pool that the volume belongs to. If the volume has more than one copy, these fields display many.

mdisk_grp_name
Indicates the name of the storage pool that the volume belongs to. If the volume has more than one copy, these fields display many.

type Indicates the virtualization type of the volume. The value can be striped, seq, image or many. The value many indicates that the volume has more than one copy, which can have different virtualization types.

capacity
1 Indicates the virtual capacity of the volume that is the size of the volume as seen by the host.

formatted
Indicates whether the volume was formatted when it was created. The value can be yes or no.

formatting
Indicates whether the volume is formatting. The value can be yes or no.

mdisk_id
Indicates the MDisk ID that is used for sequential and image mode volumes. If the volume has more than one copy, these fields display many.

mdisk_name
Indicates the MDisk name that is used for sequential and image mode volumes. If the volume has more than one copy, these fields display many.

FC_id Indicates the ID of the FlashCopy mapping that the volume belongs to. The value many indicates that the volume belongs to more than one FlashCopy mapping.

FC_name
Indicates the name of the FlashCopy mapping that the volume belongs to. The value many indicates that the volume belongs to more than one FlashCopy mapping.

RC_id Indicates the ID of the remote copy relationship that the volume belongs to. The value must be numerical.

RC_name
Indicates the name of the remote copy relationship that the volume belongs to.

vdisk_UID
Indicates the UID of the volume.

throttle_ID
Indicates the ID for the throttle object. The value is a numeric string in the range 0 - 10241 (or is blank if no throttle is configured).

throttle_name
Indicates the name of the throttle object. The value is an alphanumeric string in the range 1 - 63 characters or blank if no throttle is specified.

IOPs_limit
Indicates the IOPs limit that is configured for the volume. The value is a numeric string or blank if no limit is specified.

bandwidth_limit_MB
Indicates the bandwidth limit configured (in MB) for the volume. The value is a numeric string or is blank if no limit is configured.

preferred_node_id
Indicates the node that processes the I/O data.

Remember: This value must be numeric. (The value is zero if no node is configured in the I/O group that contains the preferred node.)

fast_write_state
Indicates the cache state for the volume or volume copy. The value can be `empty`, `not_empty`, `corrupt`, or `repairing`. A cache state of `corrupt` indicates that the volume or volume copy requires repairing or recovery by using either the **`recovervdisk`** or **`repairvdiskcopy`** command.

cache Indicates the cache mode of the volume. The value can be `readonly`, `readwrite`, or `none`.

udid Indicates the unit number for the volume. Only OpenVMS hosts require a unit number.

fc_map_count
Indicates the number of FlashCopy mappings that the volume belongs to.

sync_rate
Indicates the rate for synchronization for mirrored copies.

se_copy_count
Indicates the number of thin-provisioned copies.

Remember: This value represents only thin-provisioned copies and is not used for compressed volume copies.

filesystem
Expressed as a value string (long object name with a maximum of 63 characters), indicates the full name for file system that owns this volume; otherwise, it is blank.

mirror_write_priority
Indicates the mirror write algorithm priority that is used if the volume is mirrored.

RC_change
Indicates whether a volume is a change volume of a remote copy relationship.

compressed_copy_count
Indicates the number of compressed volume copies.

access_IO_group_count
Indicates the number of I/O groups in the volume access set.

The command returns values for the following volume copy attributes:

copy_id
Indicates a system-assigned identifier for the volume copy. The value can be `0` or `1`.

status Indicates the status. The value can be `online`, `offline`, `degraded`, or `deleting`.

sync Indicates whether the volume copy is synchronized.

auto_delete
Indicates that the primary copy is deleted after the secondary copy is synchronized. The values are `yes` or `no`.

primary
Indicates whether the volume copy is the primary copy. A volume has exactly one primary copy. The value can be `Yes` or `No`.

mdiskgrp_id
Indicates the ID of the storage pool that the volume copy belongs to.

mdiskgrp_name
Indicates the name of the storage pool that the volume copy belongs to.

type Indicates the virtualization type of the volume. The value can be `striped`, `seq`, or `image`.

mdisk_id
Indicates the MDisk ID that is used for sequential and image mode volumes.

mdisk_name
Indicates the MDisk name that is used for sequential and image mode volumes.

used_capacity
Indicates the portion of `real_capacity` that is being used to store data. For non-thin-provisioned copies, this value is the same as the volume capacity. If the volume copy is thin-provisioned, the
1 value increases from zero to the `real_capacity` value as more of the volume is written to. This
1 field is blank for volume copies that are thin-provisioned or compressed volume copies in a data
1 reduction pool.

real_capacity
Indicates the amount of physical storage that is allocated from a storage pool to this volume copy. If the volume copy is not thin-provisioned, the value is the same as the volume capacity. If the
1 volume copy is thin-provisioned, the value can be different. This field is blank for volume copies
1 that are thin-provisioned or compressed volume copies in a data reduction pool.

free_capacity
Indicates the difference between the `real_capacity` and `used_capacity` values. This field is blank for storage pools that are not thin-provisioned or compressed volume copies in a data reduction pool.

overallocation
Expressed as a percentage of the volume capacity, indicates the ratio of capacity to `real_capacity` values. This value is always `100` for non-thin-provisioned or compressed volumes.

Remember:

1 - This field is blank for volume copies that are thin-provisioned or compressed volume copies in
1 a data reduction pool.
- This value cannot be blank for compressed volume copies.

1 This field is blank for volume copies that are thin-provisioned or compressed volume copies in a
1 data reduction pool.

autoexpand
Indicates whether **`autoexpand`** is enabled on a thin-provisioned volume. The value can be `on` or `off`.

Remember: This value cannot be blank for compressed copies.

warning
Expressed as a percentage of the volume capacity, this value indicates a warning for thin-provisioned or compressed volume copies. A warning is generated when the ratio of `used_capacity` to volume capacity reaches the specified level.

Remember:

- 1 This field is blank for volume copies that are thin-provisioned or compressed volume copies in
  1 a data reduction pool.
- This value cannot be blank for compressed volume copies.

grainsize

For thin-provisioned volume copies, indicates the grain size that is chosen for the volume copy when it was created.

1 **Remember:** This value is always blank for compressed volume copies in regular storage pools.

se_copy

Indicates whether the copy is thin-provisioned.

Remember: This value is `yes` for thin-provisioned copies and `no` for compressed volume copies.

easy_tier

This value is set by the user and indicates whether Easy Tier is permitted to manage the pool.

Note:

1. If easy_tier is `on`, then easy_tier_status can take on any value.
2. If easy_tier is `off`, then easy_tier_status is `measured` or `inactive`.

easy_tier_status

Indicates which Easy Tier functions are active for the volume copy:

- `active` indicates that a pool is being managed by Easy Tier to provide tier management performance-based pool balancing. For example, extents of this volume copy can be moved for performance (automatic data placement).
- `inactive` indicates that no Easy Tier function is active.
- `measured` indicates that statistics are being gathered for this volume copy, but no extents are moved.
- `balanced` indicates that a pool is being managed by Easy Tier to provide performance-based pool balancing (for example, extents can be moved).

This table displays possible values and related information for `easy_tier_status`:

Table 105. Easy Tier status values. Easy Tier status values

Storage pool Easy Tier setting	Number of tiers in the storage pool	Volume copy Easy Tier setting	Volume copy Easy Tier status
Off	One	Off	inactive (see note 1 on page 686)
Off	One	On	inactive (see note 1 on page 686)
Off	Two	Off	inactive (see note 1 on page 686)
Off	Two	On	inactive (see note 1 on page 686)
Measure	One	Off	measured (see note 2 on page 686)
Measure	One	On	measured (see note 2 on page 686)
Measure	Two	Off	measured (see note 2 on page 686)
Measure	Two	On	measured (see note 2 on page 686)

Table 105. Easy Tier status values (continued). Easy Tier status values

Storage pool Easy Tier setting	Number of tiers in the storage pool	Volume copy Easy Tier setting	Volume copy Easy Tier status
Auto	One	Off	measured (see note 2)
Auto	One	On	balanced (see note 3)
Auto	Two	Off	measured (see note 2)
Auto	Two	On	(see note 4)
On	One	Off	measured (see note 2)
On	One	On	balanced (see note 3)
On	Two	Off	measured (see note 2)
On	Two	On	active (see note 4)

Notes:

1. When the volume copy status is `inactive`, no Easy Tier functions are enabled for that volume copy.
2. When the volume copy status is `measured`, the Easy Tier function collects usage statistics for the volume but automatic data placement is not active.
3. When the volume copy status is `balanced`, the Easy Tier function enables performance-based pool balancing for that volume copy.
4. When the volume copy status is `active`, the Easy Tier function operates in automatic data placement mode for that volume.

If the volume copy is in image or sequential mode or is being migrated, the volume copy Easy Tier status is `measured` instead of `active`.

The default Easy Tier setting for a storage pool is `auto`, and the default Easy Tier setting for a volume copy is `on`. This means that Easy Tier functions except pool performance balancing are disabled for storage pools with a single tier, and that automatic data placement mode are enabled for all striped volume copies in a storage pool with two or more tiers.

tier The tier information being reported:

- `tier0_flash`
- `tier1_flash`
- `tier_enterprise`
- `tier_nearline`

tier_capacity
The total MDisk capacity that is assigned to the volume in the tier.

Note: For thin-provisioned copies, the capacity by tier is the real capacity.

compressed_copy
Indicates whether the volume copy is compressed.

uncompressed_used_capacity
1 For compressed volume copies, indicates the amount of data written to the volume copy before
1 compression. This field is blank for volume copies in a data reduction storage pool.

used_capacity_before_reduction
1 Indicates the total amount of data written to a thin-provisioned or compressed volume copy in a
1 data reduction storage pool before data reduction occurs. This field is blank for fully allocated
1 volume copies and volume copies not in a data reduction pool.

last_access_time
Indicates the time (`YYMMDDHHMMSS`) the volume last received Small Computer System Interface (SCSI) commands from any mapped host.

parent_mdisk_grp_id

Indicates the physical storage pool ID that the volume extents are allocated from. This value is a numeric string (in the range 0 - 127 characters) or blank.

parent_mdisk_grp_name

Indicates the physical storage pool name that the volume extents are allocated from. This value is an alphanumeric string (in the range 1 - 63 characters) or blank.

owner_type

Indicates the type of owning component or object (such as a file system). The values are:

- `filesystem`
- `host_integration_metadata`
- `vvol`
- `none`

The value is `none` if no owner is assigned.

owner_id

Indicates an identification number for the owning object. This value must be a numeric character unless there is no owning object (in which case it is blank).

owner_name

Indicates the name for owning object that owns this volume. The value must be a set of up to 63 alphanumeric characters, but is blank if there is no owning object.

encrypt

Indicates whether all copies of a volume reside in MDisk groups (storage pools) that are reported as encrypting. This means that either one of the following things apply:

- The copies reside in a storage pool that has an encryption key.
- All MDisks in the group are self-encrypting or encrypted for RAID.

The values are `yes` and `no`.

volume_id

Indicates the volume ID (for a high availability volume). This ID must be a numerical value. For a basic or stretched volume, `volume_ID` has the same value as `id`. For a HyperSwap volume (involved in an `active-active` relationship), the volume ID is the same as the master volume.

volume_name

Indicates the volume name (for a high availability volume). This value must be an alphanumeric string that contains up to 63 characters. For a basic or stretched volume, `volume_name` has the same value as `name`. For a HyperSwap volume (involved in an `active-active` relationship), the volume name is the same as the master volume.

function

Indicates the function of the volume in the remote copy relationship. Remote copy includes Metro Mirror, Global Mirror, and HyperSwap.

The values are:

- `master`, which indicates a master volume in a remote copy relationship.
- `aux`, which indicates an auxiliary volume in a remote copy relationship.
- `master_change`, which indicates a change volume for a master volume in a remote copy relationship.
- `aux_change`, which indicates a change volume for an auxiliary volume in a remote copy relationship.
- Blank, which indicates that the volume is not in any remote copy relationship.

volume_group_id

Indicates a volume group ID for a volume group that a volume belongs to. The value must be a number.

volume_group_name

Indicates the volume group name for a volume group that a volume belongs to. The value must be an alphanumeric string.

cloud_backup_enabled

Indicates whether the cloud snapshot feature is enabled for the specified volume. The values are `yes` or `no`.

cloud_account_id

Indicates the cloud account ID. The value must be a number.

cloud_account_name

Indicates the cloud account name. The value must be an alphanumeric string.

backup_status

Indicates whether a new cloud snapshot can be started. If a backup is in progress, the status of the backup operation is given. The values are:

- `off`
- `ready`
- `copying`
- `copying_error`
- `not_ready`

last_backup_time

Indicates the time of the most recent backup or snapshot for the specified volume. The value must be in `YYMMDDHHMMSS` format (or blank).

restore_status

Indicates whether a restore can be performed for the volume. If a restore is in progress, the status of the restore operation is given. The values are:

- `none`
- `available`
- `restoring`
- `restoring_error`
- `committed`
- `committing`
- `committing_error`

backup_grain_size

Indicates the grain size for the volume mappings that are used for the cloud snapshot function. The value is blank if cloud snapshot is not enabled.

Note: This sizing does not reflect the size of the grains that are stored in the cloud (which are fixed at 256 KB).

1 **deduplicated_copy_count**

1 Indicates the number of data deduplicated volume copies.

1 **deduplicated_copy**

1 Indicates whether the volume copy is data deduplicated. The values are:

1 - `yes`

1 - `no`

A detailed invocation example for a volume

```
lsvdisk -delim : vv45
```

The following output is displayed:

```
name:vv45
IO_group_id:0
IO_group_name:io_grp0
status:online
mdisk_grp_id:0
mdisk_grp_name:Group0
capacity:1000.00MB
type:striped
formatted:no
formatting:yes
mdisk_id:
mdisk_name:
FC_id:
FC_name:
RC_id:
RC_name:
vdisk_UID:60050768019B82328000000000000010
preferred_node_id:2
fast_write_state:empty
cache:readwrite
udid:
fc_map_count:0
sync_rate:50
copy_count:1
se_copy_count:0
filesystem:
mirror_write_priority:redundancy
RC_change:no
compressed_copy_count:0
access_IO_group_count:1
parent_mdisk_grp_id:5
parent_mdisk_grp_name:p5
encrypt:yes
volume_id:0
volume_name:homer0
function:aux
owner_type filesystem
owner_id 2
owner_name myfilesystem2
copy_id:0
status:online
sync:yes
auto_delete:yes
primary:yes
mdisk_grp_id:0
mdisk_grp_name:Group0
type:striped
mdisk_id:
mdisk_name:
fast_write_state:empty
used_capacity:1000.00MB
real_capacity:1000.00MB
free_capacity:0.00MB
overallocation:100
autoexpand:
warning:
grainsize:
se_copy:no
easy_tier:on
easy_tier_status:inactive

tier:tier0_flash
```

```
tier_capacity:1.63TB
tier:tier1_flash
tier_capacity:1.63TB
tier:tier_enterprise
tier_capacity:0.00MB
tier:tier_nearline
tier_capacity:0.00MB
block_size:4096
compressed_copy:no
uncompressed_used_capacity:1000.00MB
used_capacity_before_reduction
last_access_time:140604171325
throttle_id:1
throttle_name:lcyfoxes_1
IOPs_limit:25000
bandwidth_limit_MB:500
volume_group_id:1
volume_group_name:ZlaIbra2
cloud_backup_enabled:no
cloud_account_id:
cloud_account_name:
backup_status:off
last_backup_time:
restore_status:available
backup_grain_size:
used_capacity_before_reduction
```

A concise invocation example

```
lsvdisk -delim :
```

The following output is displayed:

```
id:name:IO_group_id:IO_group_name:status:mdisk_grp_id:mdisk_grp_name:capacity:type:
FC_id:FC_name:RC_id:RC_name:vdisk_UID:fc_map_count:copy_count:
fast_write_state:se_copy_count:RC_change:compressed_copy_count:volume_id:volume_name:funtion
0:vdisk0:0:io_grp0:degraded:0:mdiskgrp0:10.00GB:striped:::::60050768018300003000000000000000:0:1:empty:0:no:0:1:VDisk1:aux_
```

A detailed invocation example

```
lsvdisk -delim : vv1
```

The following output is displayed:

```
id:0
name:vv1
IO_group_id:0
IO_group_name:io_grp0
status:degraded
mdisk_grp_id:many
mdisk_grp_name:many
capacity:16.00GB
type:many
formatted:no
formatting:yes
mdisk_id:many
mdisk_name:many
FC_id:
FC_name:
RC_id:
RC_name:
vdisk_UID:00000000000000AB:6005076801CF003F2800000000000000
preferred_node_id:1
fast_write_state:empty
cache:readwrite
udid:1234
fcmap_count:0
sync_rate:25
```

```
copy_count:2
se_copy_count:1filesystem:
mirror_write_priority:redundancy
RC_change:no
compressed_copy_count:0
access_IO_group_count:1
parent_mdisk_grp_id:5
parent_mdisk_grp_name:p5
encrypt:yes
volume_id:1
volume_name:slayer1
function:aux
owner_type filesystem
owner_id 2
owner_name myfilesystem2
copy_id:0
status:online
sync:yes
auto_delete:yes
primary:yes
mdisk_grp:1
mdisk_grp_name:mdisk_group_1
type:striped
mdisk_id:
mdisk_name:
fast_write_state:corrupt
used_capacity:8.00GB
real_capacity:8.00GB
free_capacity:6.00GB
overallocation:100
autoexpand:off
warning:
grainsize:
se_copy:no
easy_tier:off
easy_tier_status:inactive

tier:tier0_flash
tier_capacity:1.63TB
tier:tier1_flash
tier_capacity:1.63TB
tier:tier_enterprise
tier_capacity:0.00MB
tier:tier_nearline
tier_capacity:0.00MB
block_size:4096
compressed_copy:no
uncompressed_used_capacity:1000.00MB
used_capacity_before_reduction
copy_id:1
status:offline
sync:no
primary:no
mdisk_grp:2
mdisk_grp_name:mdisk_group_2
type:striped
mdisk_id:
mdisk_name:
fast_write_state:not_empty
used_capacity:2.00GB
real_capacity:4.00GB
free_capacity:2.00GB
overallocation:400
autoexpand:on
warning:20
grainsize:256
se_copy:yes
```

```
easy_tier:on
easy_tier_status:active

tier:tier0_flash
tier_capacity:1.63TB
tier:tier1_flash
tier_capacity:1.63TB
tier:tier_enterprise
tier_capacity:0.00MB
tier:tier_nearline
tier_capacity:0.00MB
block_size:4096
compressed_copy:no
uncompressed_used_capacity:1000.00MB
used_capacity_before_reduction
last_access_time 140604171325
parent_mdisk_grp_id:5
parent_mdisk_grp_name:p5
throttle_id:1
throttle_name:lcyfoxes_1
IOPs_limit:25000
bandwidth_limit_MB:500
volume_group_id:1
volume_group_name:ZlaIbra2
cloud_backup_enabled:no
cloud_account_id:
cloud_account_name:
backup_status:off
last_backup_time:
restore_status:available
backup_grain_size:
used_capacity_before_reduction
```

An invocation example

```
lsvdisk -delim : vv2
```

The following output is displayed:

```
id:0
name:vv2
IO_group_id:0
IO_group_name:io_grp0
status:degraded
mdisk_grp_id:many
mdisk_grp_name:many
capacity:16.00GB
type:many
formatted:no
formatting:yes
mdisk_id:many
mdisk_name:many
FC_id:
FC_name:
RC_id:
RC_name:
vdisk_UID:00000000000000AB:6005076801CF003F2800000000000000
preferred_node_id:1
fast_write_state:empty
cache:readwrite
udid:1234
fc_map_count:0
sync_rate:25
copy_count:2
se_copy_count:2
filesystem:
mirror_write_priority:latency
RC_change:no
```

```
compressed_copy_count:0
parent_mdisk_grp_id:5
parent_mdisk_grp_name:p5
encrypt:yes
volume_id:0
volume_name:vv2
function:master
copy_id:0
status:online
sync:yes
auto_delete:yes
primary:yes
mdisk_grp_id:1
mdisk_grp_name:mdisk_group_1
type:striped
mdisk_id:
mdisk_name:
fast_write_state:empty
used_capacity:2.00GB
real_capacity:8.00GB
free_capacity:6.00GB
overallocation:200
autoexpand:on
warning:25
grainsize:256
se_copy:yes
easy_tier:off
easy_tier_status:inactive

block_size:4096
compressed_copy:no
uncompressed_used_capacity:2.00GB
used_capacity_before_reduction
tier tier0_flash
tier_capacity:1.63TB
tier:tier1_flash
tier_capacity:1.63TB
tier:tier_enterprise
tier_capacity:0.00MB
tier:tier_nearline
tier_capacity:0.00MB
block_size:4096
compressed_copy:no
uncompressed_used_capacity:2.00GB
used_capacity_before_reduction
parent_mdisk_grp_id:5
parent_mdisk_grp_name:p5
copy_id:1
status:offline
sync:no
primary:no
mdisk_grp_id:2
mdisk_grp_name:mdisk_group_2
type:striped
mdisk_id:
mdisk_name:
fast_write_state:not_empty
used_capacity:2.00GB
real_capacity:4.00GB
free_capacity:2.00GB
overallocation:400
autoexpand:on
warning:20
grainsize:256
se_copy:yes
easy_tier:off
easy_tier_status:inactive
```

```
block_size:4096
compressed_copy:no
uncompressed_used_capacity:2.00GB
used_capacity_before_reduction
tier tier0_flash
tier_capacity:1.63TB
tier:tier1_flash
tier_capacity:1.63TB
tier:tier_enterprise
tier_capacity:0.00MB
tier:tier_nearline
tier_capacity:0.00MB
block_size:4096
compressed_copy:no
uncompressed_used_capacity:2.00GB
used_capacity_before_reduction
last_access_time 140604171325
parent_mdisk_grp_id:5
parent_mdisk_grp_name:p5
throttle_id:1
throttle_name:lcyfoxes_1
IOPs_limit:25000
bandwidth_limit_MB:500
volume_group_id:1
volume_group_name:ZlaIbra2
cloud_backup_enabled:no
cloud_account_id:
cloud_account_name:
backup_status:off
last_backup_time:
restore_status:available
backup_grain_size:
used_capacity_before_reduction
```

1 A detailed invocation example for a new style volume

1 `lsvdisk -delim : Volume0`

1 The following output is displayed:

```
1 id 0
1 name Volume0
1 ...
1 deduplicated_copy_count 1
1
1 copy_id 0
1 ...
1 deduplicated_copy yes
```

lsvdiskaccess

Use the **lsvdiskaccess** command to display a list of all input/output (I/O) groups in the volume access set.

Syntax

```
►►── lsvdiskaccess ──┬─────────────────────────────────┬──┬───────────┬── ──────────────────────────►
                     └─ -filtervalue ── attribute_value ┘  └─ -nohdr ─┘
```

```
►─┬──────────────────────┬─ ─┬────────────────┬─ ─┬──────────────┬────────────────────────────►◄
  └─ -delim ── delimiter ─┘   └─ -filtervalue? ─┘   ├─ vdisk_id ───┤
                                                    └─ vdisk_name ─┘
```

Parameters

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards:

- The wildcard character is the asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, enclose the filter entry within double quotation marks (""):
  `lsvdiskaccess -filtervalue "IO_group_name=io*"`

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-filtervalue?
(Optional) Displays the valid filter attributes for the **-filtervalue** parameter:

- IO_group_id
- IO_group_name

vdisk_id | *vdisk_name*
(Optional) Specifies the volume for which to list access I/O groups.

Description

The **lsvdiskaccess** command lists the I/O groups in a volume access set. An accessible volume in an I/O group does not indicate that the volume is mapped to any hosts. There is a detailed view and concise view, but the detailed view does not contain more information than the concise view.

This command returns values for the following volume attributes:

VDisk_id
Identifies the volume ID.

VDisk_name
Identifies the volume name.

IO_group_id
Identifies an I/O group ID in the volume access set.

IO_group_name
Identifies an I/O group name in the volume access set.

A detailed invocation example

```
lsvdiskaccess 0
```

The resulting output:

```
vdisk_id vdisk_name IO_group_id IO_group_name
0        vdisk0     0           io_grp0
0        vdisk0     1           io_grp1
0        vdisk0     2           io_grp2
```

A concise invocation example

```
lsvdiskaccess
```

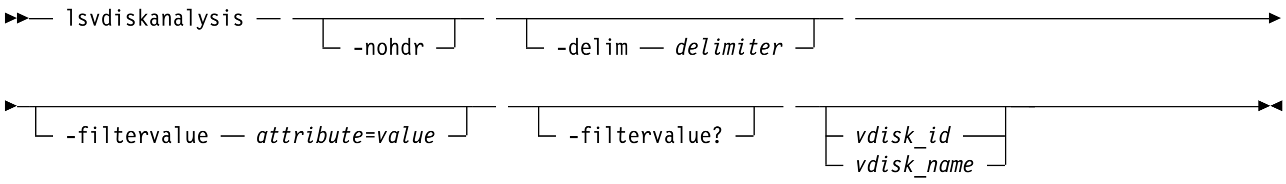
The resulting output:

```
vdisk_id vdisk_name IO_group_id IO_group_name
0        vdisk0     0           io_grp0
0        vdisk0     1           io_grp1
0        vdisk0     2           io_grp2
3        vdisk3     1           io_grp1
7        vdisk7     0           io_grp0
7        vdisk7     2           io_grp2
```

lsvdiskanalysis

Use the **lsvdiskanalysis** command to display information for thin provisioning and compression estimation analysis report for a single volume or multiple volumes.

Syntax



Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim : on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-filtervalue *attribute=value*

(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards with the SAN Volume Controller CLI:

- The wildcard character is the asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, enclose the filter entry within double quotation marks (""), as follows:

```
lsvdiskanalysis -filtervalue "usergrp_name=md*"
```

-filtervalue?

(Optional) Displays the valid filter attributes for the **-filtervalue***attribute=value* parameter:

- id
- name
- state

vdisk_id | *vdisk_name*

(Optional) Specifies the volume (by ID or name) to display compression estimation status for.

Description

This command displays information for thin provisioning and compression estimation analysis report for a single volume or multiple volumes.

Table 106 provides the attribute values that can be displayed as output view data.

Table 106. lsvdiskanalysis output

Attribute	Description
id	Indicates the ID (by number) of the object.
analysis_state	Indicates one of the following values: • `idle` indicates that the volume was never analyzed. • `scheduled` indicates that the volume is scheduled for analysis (the analysis starts based on ascending volume IDs). • `active` indicates that the volume is being analyzed. • `estimated` indicates that the volume was analyzed and the analysis results reflect the estimated savings from thin provisioning and compression. • `sparse` indicates that the volume was analyzed but not enough samples of nonzero data were found. • `cancelling` indicates the analysis is undergoing but there is a request to cancel it, and the analysis did not end.
started_time	Indicates the date and time that the analysis started, which helps determine how long an estimate takes. The value must be an alphanumeric data string or be empty if analysis never started.
analysis_time	Indicates the date and time that the analysis ends to help determine how current the results are. If you cancel analysis, the value for the time is invalid (`0`, and it is not displayed). While active, time is invalid as well and does not reflect expected completion time. The value must be an alphanumeric data string or be empty if analysis never occurred.
capacity	Indicates the virtual capacity (host size) of the volume.
thin_size	Indicates the estimated size of the data without zero portions (thin-provisioned size).
thin_savings	Indicates how much data is expected to be saved if it is a thin-provisioned volume.

Table 106. lsvdiskanalysis output (continued)

Attribute	Description
thin_savings_ratio	Indicates the percent of data that is saved by thin-provisioned. The number must be a percentage.
compressed_size	Indicates the estimated size of any nonzero data after compression completes.
compression_savings	Indicates how much data to expect to save if the volume is a compressed volume.
compression_savings_ratio	Indicates the amount of data that is saved by compression. The number must be a percentage.
total_savings	Indicates how much data to expect to save by converting a volume to a compressed volume.
total_savings_ratio	Indicates the amount of data that is saved by compression based on the overall volume capacity, which includes the thin nature of compressed volumes. The number must be a percentage.
accuracy	Indicates the accuracy estimation. The number must be a percentage.

A concise invocation example

```
lsvdiskanalysis
```

The detailed resulting output:

```
id name state     analysis_time capacity thin_size thin_savings thin_savings_ratio compressed_size compression_savings compression_savings_ratio total_savings total_savings_ratio accu
0  ben0 idle                    1.00GB   0.00MB    0.00MB       0                  0.00MB          0.00MB              0                         0.00MB        0                   0
1  ben1 idle                    1.00GB   0.00MB    0.00MB       0                  0.00MB          0.00MB              0                         0.00MB        0                   0
2  ben2 active                  1.00GB   0.00MB    0.00MB       0                  0.00MB          0.00MB              0                         0.00MB        0                   0
3  ben3 idle                    1.00GB   0.00MB    0.00MB       0                  0.00MB          0.00MB              0                         0.00MB        0                   0
4  ben4 idle                    1.00GB   0.00MB    0.00MB       0                  0.00MB          0.00MB              0                         0.00MB        0                   0
5  ben5 idle                    1.00GB   0.00MB    0.00MB       0                  0.00MB          0.00MB              0                         0.00MB        0                   0
6  ben6 estimated 150608135456  1.00GB   62.18MB   961.82MB     93.92              12.23MB         49.95MB             80.33                     1011.77MB     98.80               4.97
7  ben7 scheduled               1.00GB   0.00MB    0.00MB       0                  0.00MB          0.00MB              0                         0.00MB        0                   0
8  ben8 idle                    1.00GB   0.00MB    0.00MB       0                  0.00MB          0.00MB              0                         0.00MB        0                   0
```

A detailed invocation example

```
lsvdiskanalysis
```

The detailed resulting output:

```
id 6
name ben6
state estimated
started_time 150608135446
analysis_time 150608135456
capacity 1.00GB
thin_size 62.18MB
thin_savings 961.82MB
thin_savings_ratio 93.92
compressed_size 12.23MB
compression_savings 49.95MB
compression_savings_ratio 80.33
total_savings 1011.77MB
total_savings_ratio 98.80
accuracy 4.97
```

lsvdiskanalysisprogress

Use the **lsvdiskanalysisprogress** command to display information about the space analysis progress for an entire clustered system (system).

Syntax

```
►►── lsvdiskanalysisprogress ── ┬─────────┬─ ┬──────────────────────┬─ ─────────────────►◄
                                └─ -nohdr ┘  └─ -delim ── delimiter ┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (**:**) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

Description

This command displays information about the space analysis progress for an entire system.

Table 107 provides the attribute values that can be displayed as output view data.

Table 107. lsvdiskanalysisprogress output

Attribute	Description
vdisk_count	Indicates the number of volumes on this system.
pending_analysis	Indicates the number of volumes that belong to this system and: • Scheduled a free space analysis • Have an active free space analysis • Are canceling free space analysis
estimated_completion_time	Indicates the estimated time at which analysis is expected to end. It is calculated based on number of scheduled volumes that are multiplied by 1 minute (there is no extrapolation from actual analysis duration). Estimated completion time does not consider volumes that are offline and displays estimated completion time as if the volumes are online.

A concise invocation example that shows progress for a system with some scheduled disks

```
lsvdiskanalysisprogress
```

The detailed resulting output:

```
vdisk_count pending_analysis estimated_completion_time
15          10               20150523135200
```

A concise invocation example that shows progress for a system with no scheduled disks

```
lsvdiskanalysisprogress
```

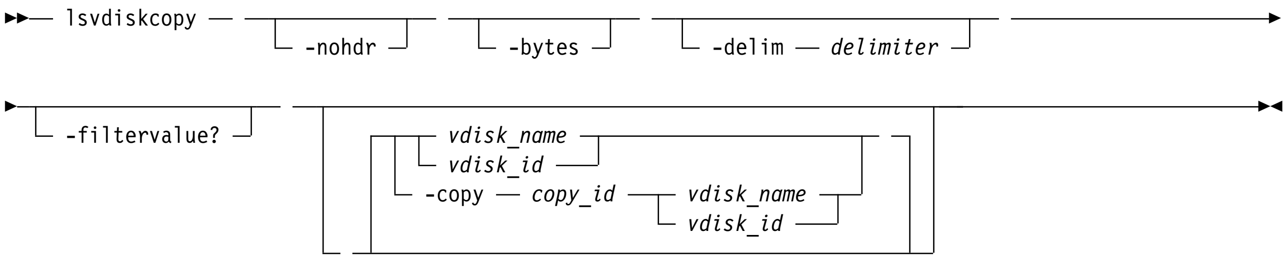
The detailed resulting output:

```
vdisk_count pending_analysis estimated_completion_time
15          0
```

lsvdiskcopy

Use the **lsvdiskcopy** command to list volume copy information.

Syntax



Parameters

-nohdr

(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If no data exists to be displayed, headings are not displayed.

-bytes

(Optional) Displays all capacities as bytes. Capacity values that are displayed in units other than bytes might be rounded.

-delim *delimiter*

(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim : on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-copy *copy_id*

(Optional) Specifies the volume copy to list information for. You must specify a *vdisk_name* | *vdisk_id* value with this parameter.

-filtervalue?

(Optional) Displays a list of valid filter attributes. The following filters for the **lsvdiskcopy** command are valid:

- primary
- status
- sync
- mdisk_grp_id
- mdisk_grp_name
- type
- easy_tier
- easy_tier_status

1 - deduplicated_copy

vdisk_name | *vdisk_id*
(Optional) Specifies the volume to list copy information for. You must specify this parameter last on the command line. If you specify a *vdisk_name* | *vdisk_id* value only, all copies for the volume are listed.

Description

The **lsvdiskcopy** command lists information for volume copies. If you specify the command with no parameters, all volumes and copies in the clustered system are listed.

The command returns values for the following volume copy attributes:

copy_id
Specifies a system-assigned identifier for the volume copy. The value can be 0 or 1.

status Indicates the status. The value can be online, offline, or deleting. A copy is offline if all nodes cannot access the storage pool that contains the copy.

sync Indicates whether the volume copy is synchronized.

auto_delete
Indicates that the primary copy is deleted after the secondary copy is synchronized. The values are yes or no.

primary
Indicates whether the volume copy is the primary copy. A volume has exactly one primary copy. The value can be yes or no.

mdiskgrp_id
Indicates the ID of the storage pool that the volume copy belongs to.

mdiskgrp_name
Indicates the name of the storage pool that the volume copy belongs to.

type Indicates the virtualization type of the volume. The value can be striped, sequential, or image.

mdisk_id
Indicates the MDisk ID that is used for sequential and image mode volumes.

mdisk_name
Indicates the MDisk name that is used for sequential and image mode volumes.

fast_write_state
Indicates the cache state of the volume copy. The value can be empty, not_empty, corrupt, or repairing. The value is always empty for non-thin-provisioned copies. A cache state of corrupt indicates that the volume is thin-provisioned and requires repair that is initiated by a **recovervdisk** command or the **repairsevdiskcopy** command.

used_capacity
Indicates the portion of real_capacity that is being used to store data. For non-thin-provisioned copies, this value is the same as the volume capacity. If the volume copy is thin-provisioned, the value increases from zero to the real_capacity value as more of the volume is written to.

Remember:

- This value is the same as the volume capacity value for fully allocated copies.
1 - This field is blank for volume copies that are thin-provisioned or compressed volume copies in
1 a data reduction pool.

real_capacity
Indicates the amount of physical storage that is allocated from a storage pool to this volume copy.

If the volume copy is not thin-provisioned, the value is the same as the volume capacity. If the volume copy is thin-provisioned, the value can be different.

Remember:

- This value is the same as the volume capacity value for fully allocated copies.
1 - This field is blank for volume copies that are thin-provisioned or compressed volume copies in
1 a data reduction pool.

free_capacity

Indicates the difference between the `real_capacity` and `used_capacity` values.

Remember:

- This value is `0` for fully allocated copies.
1 - This field is blank for volume copies that are thin-provisioned or compressed volume copies in
1 a data reduction pool.

Remember: This value is zero for fully allocated copies.

overallocation

Expressed as a percentage, indicates the ratio of volume capacity to `real_capacity` values. This value is always `100` for non-thin-provisioned volumes.

Remember: This field is blank for storage pools that are not thin-provisioned or compressed volume copies in a data reduction pool.

autoexpand

Indicates whether `autoexpand` is enabled on a thin-provisioned volume. The value can be `on` or `off`.

warning

Expressed as a percentage of the volume capacity for thin-provisioned or compressed volume copies, indicates that a warning is generated when the ratio of `used_capacity` to volume capacity reaches the specified level.

Remember: This field is blank for storage pools that are not thin-provisioned or compressed volume copies in a data reduction pool.

grainsize

For thin-provisioned volume copies, indicates the grain size that is chosen for the volume copy when it was created.

1 **Remember:** This value is always blank for compressed volume copies in regular storage pools.

se_copy

Specifies whether the copy is thin-provisioned.

easy_tier

Indicates whether Easy Tier is permitted to manage the pool.

Note:

1. If easy_tier is `on`, then easy_tier_status can take on any value.
2. If easy_tier is `off`, then easy_tier_status is `measured` or `inactive`.

easy_tier_status

Indicates which Easy Tier functions are active for the volume copy:

- `active` indicates that a pool is being managed by Easy Tier to provide tier management performance-based pool balancing. For example, extents of this volume copy can be moved for performance (automatic data placement).

- inactive indicates that no Easy Tier function is active.
- balanced indicates that a pool is being managed by Easy Tier to provide performance-based pool balancing (for example, extents can be moved).
- measured indicates that statistics are being gathered for this volume copy, but no extents are moved.

Table 108. Easy Tier setting for storage pools and volumes

Storage pool Easy Tier setting	Number of tiers in the storage pool	Volume copy Easy Tier setting	Volume copy Easy Tier status
Off	One	Off	inactive (see note 1)
Off	One	On	inactive (see note 1)
Off	Two	Off	inactive (see note 1)
Off	Two	On	inactive (see note 1)
Measure	One	Off	measured (see note 2)
Measure	One	On	measured (see note 2)
Measure	Two	Off	measured (see note 2)
Measure	Two	On	measured (see note 2)
Auto	One	Off	measured (see note 2)
Auto	One	On	balanced (see note 3)
Auto	Two	Off	measured (see note 2)
Auto	Two	On	active (see note 4)
On	One	Off	measured (see note 2)
On	One	On	balanced (see note 3)
On	Two	Off	measured (see note 2)
On	Two	On	active (see note 4)

Notes:

1. When the volume copy status is inactive, no Easy Tier functions are enabled for that volume copy.
2. When the volume copy status is measured, the Easy Tier function collects usage statistics for the volume but automatic data placement is not active.
3. When the volume copy status is balanced, the Easy Tier function enables performance-based pool balancing for that volume copy.
4. When the volume copy status is active, the Easy Tier function operates in automatic data placement mode for that volume.

If the volume copy is in image or sequential mode or is being migrated, then the volume copy Easy Tier status is `measured` instead of `active`.

The default Easy Tier setting for a storage pool is `auto`, and the default Easy Tier setting for a volume copy is `on`. If the setting is `on`, it means that Easy Tier functions except pool performance balancing are disabled for storage pools with a single tier, and that automatic data placement mode is enabled for all striped volume copies in a storage pool with two or more tiers.

tier Indicates which tier information is being reported:

- `tier0_flash`
- `tier1_flash`
- `tier_enterprise`
- `tier_nearline`

tier_capacity
Indicates the total MDisk capacity that is assigned to the volume in the tier.

Note: For thin-provisioned copies, the capacity by tier is the real capacity.

1 **Note:** By design, **`tier_capacity`** reports blank for thin-provisioned and compressed copies in data reduction pools.

compressed_copy
Indicates whether the volume copy is compressed.

uncompressed_used_capacity
1 For compressed volume copies, indicates the amount of data written to the volume copy before
1 compression. This field is blank for volume copies in a data reduction storage pool.

used_capacity_before_reduction
1 Indicates the total amount of data written to a thin-provisioned or compressed volume copy in a
1 data reduction storage pool before data reduction occurs. This field is blank for fully allocated
1 volume copies and volume copies not in a data reduction pool.

parent_mdisk_grp_id
Indicates the physical storage pool ID that the volume extents are allocated from. This value is a numeric string (in the range 0 - 127 characters) or blank.

parent_mdisk_grp_name
Indicates the physical storage pool name that the volume extents are allocated from. This value is an alphanumeric string (in the range 1 - 63 characters) or blank.

encrypt
Indicates whether the volume and its copies are encrypted. The values are `yes` or `no`.

1 **deduplicated_copy_count**
1 Indicates the number of data deduplicated volume copies.

1 **deduplicated_copy**
1 Indicates whether the volume copy is data deduplicated. The values are:

1 - `yes`
1 - `no`

An invocation example

```
lsvdiskcopy -delim :
```

The following output is displayed:

```
vdisk_id:vdisk_name:copy_id:status:sync:primary:mdisk_grp_id:mdisk_grp_name:
capacity:type:se_copy:easy_tier:easy_tier_status:compressed_copy
0:RAM_V2:0:online:yes:yes:2:RAM_MDG2:5.00GB:striped:yes:on:inactive:yes
```

```
1:RAM_V3:0:online:yes:yes:2:RAM_MDG2:5.00GB:striped:no:on:inactive:no
2:RAM_V4:0:online:yes:yes:1:RAM_MDG3:5.00GB:striped:no:on:inactive:yes
3:RAM_V5:0:online:yes:yes:2:RAM_MDG2:5.00GB:striped:yes:on:inactive:no
3:RAM_V5:1:online:yes:no:2:RAM_MDG2:5.00GB:striped:yes:on:inactive:yes
4:RAM_V1:0:online:yes:yes:3:RAM_MDG1:5.00GB:striped:no:on:inactive:no
5:RAM_V6:0:online:yes:yes:0:RAM_MDG4:5.00GB:striped:yes:on:inactive:yes
```

An invocation example

```
lsvdiskcopy -copy 0 –delim : vv1
```

The following output is displayed:

```
vdisk_id:0
vdisk_name:vv1
capacity:16.00GB
copy_id:0
status:online
sync:yes
auto_delete:yes

primary:yes
mdisk_grp:1
mdisk_grp name:mdisk_group_1
type:striped
mdisk_id:
mdisk_name:
fast_write_state:not_empty
used_capacity:2.00GB
real_capacity:8.00GB
free_capacity:6.00GB
overallocation:200
autoexpand:on
warning:25
grainsize:256
se_copy:yes
easy_tier:on
easy_tier_status:active

tier:tier0_flash
tier_capacity:1.63TB
tier:tier1_flash
tier_capacity:1.63TB
tier:tier_enterprise
tier_capacity:
tier:tier_nearline
tier_capacity:0.00MB
tier_capacity:64.00MB

tier:tier0_flash
tier_capacity:1.63TB
tier:tier1_flash
tier_capacity:1.63TB
tier:tier_enterprise
tier_capacity:
tier:tier_nearline
tier_capacity:0.00MB
tier_capacity:7.94GB
compressed_copy:yes
uncompressed_used_capacity:1.0MB
parent_mdisk_grp_id:5
parent_mdisk_grp_name:p5
encrypt:yes
used_capacity_before_reduction
```

An invocation example

```
lsvdiskcopy -copy 0 –delim : vv1
```

The following output is displayed:

```
vdisk_id:0
vdisk_name:vv1
capacity:16.00GB
copy_id:0
status:online
sync:yes
auto_delete:yes

primary:yes
mdisk_grp_id:1
mdisk_grp_name:mdisk_group_1
type:striped
mdisk_id:
mdisk_name:
fast_write_state:empty
used_capacity:2.00GB
real_capacity:8.00GB
free_capacity:6.00GB
overallocation:200
autoexpand:on
warning:25
grainsize:256
se_copy:yes
easy_tier:off
easy_tier_status:inactive

tier:tier0_flash
tier_capacity:1.63TB
tier:tier1_flash
tier_capacity:1.63TB
tier:tier_enterprise
tier_capacity:
tier:tier_nearline
tier_capacity:0.00MB
compressed_copy:no
uncompressed_used_capcaity:8.00GB
parent_mdisk_grp_id:5
parent_mdisk_grp_name:p5
encrypt:yes
used_capacity_before_reduction
```

1 An invocation example

```
1 lsvdisk 0
```

1 The following output is displayed:

```
1 vdisk_id:0
1 vdisk_name:vv1
1 ...
1 deduplicated_copy_count:1
1 ..
1 ...
1 copy_id 0
1 ...
1 deduplicated_copy:yes
1 used_capacity_before_reduction:12.54GB
```

lsvdiskdependentmaps

Use the **lsvdiskdependentmaps** command to display all FlashCopy mappings with target volumes that are dependent upon data that is held on the specified volume.

Syntax

```
►►── lsvdiskdependentmaps ── ─┬─────────┬─ ─┬─────────────────────┬─ ─┬─ vdisk_id ───┬──────────►◄
                              └─ -nohdr ─┘  └─ -delim ─ delimiter ─┘   └─ vdisk_name ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

vdisk_id | *vdisk_name*
(Required) Specifies the name or ID of a volume.

Description

The **lsvdiskdependentmaps** command displays FlashCopy mappings that have target volumes that are dependent upon data that is held on the specified *vdisk_id* | *vdisk_name*. This data can be used to determine whether a FlashCopy mapping can be prepared. Issue the command for the target volume *vdisk_id* | *vdisk_name* of the FlashCopy mapping to be prepared. If no FlashCopy mappings are returned, the FlashCopy mapping can be prepared. Any FlashCopy mappings that are returned in the list must be stopped or be in the `idle_or_copied` state before the new FlashCopy mapping can be prepared.

A concise invocation example

```
lsvdiskdependentmaps -delim : 0
```

The concise resulting output:

```
id:name
2:fcmap2
5:fcmap5
```

lsvdiskextent

Use the **lsvdiskextent** command to list the MDisk extents that are provided for the specified volumes.

Syntax

```
►►── lsvdiskextent ── ─┬─────────────────────┬─ ─┬─────────┬─ ─┬─────────────────────┬─ ──────────►
                       └─ -copy ─ copy_id ───┘  └─ -nohdr ─┘  └─ -delim ─ delimiter ─┘
```

```
►─┬─ vdisk_name ─┬──────────────────────────────────────────────────────►◄
  └─ vdisk_id ───┘
```

Parameters

-copy *copy_id*
(Optional) Displays a list of MDisks that are members of the specified volume copy.

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If no data exists to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

vdisk_name | *vdisk_id*
(Required) Specifies one or more volume IDs or names.

Description

Note: You cannot specify this command for a thin-provisioned or compressed volume or a volume copy that is in a data reduction storage pool.

The **lsvdiskextent** command displays a list of MDisk IDs and the number of extents that each MDisk provides to the specified volumes.

Each volume is constructed from one or more MDisks. To determine the relationship between a volume and its MDisks, issue the following command:

```
lsvdiskmember vdisk_name | vdisk_id
```

where *vdisk_name* | *vdisk_id* is the name or ID of the volume. This command displays a list of MDisk IDs that make up the volume.

To determine the number of extents that are provided by each MDisk, issue the following command:

```
lsvdiskextent vdisk_name | vdisk_id
```

where *vdisk_name* | *vdisk_id* is the name or ID of the volume. This command displays a table of MDisk IDs and the corresponding number of extents that each MDisk provides as storage for the specified volume.

To determine the relationship between MDisks and volumes, issue the following command for each MDisk:

```
lsmdiskmember mdisk_name | mdisk_id
```

where *mdisk_name* | *mdisk_id* is the name or ID of the MDisk. This command displays a list of IDs that corresponds to the volumes that are using this MDisk.

To determine the relationship between MDisks and volumes, and the number of extents that are used by each volume, you must use the command-line interface. For each MDisk, issue the following command:

```
lsmdiskextent mdisk_name | mdisk_id
```

where *mdisk_name* | *mdisk_id* is the name or ID of the MDisk. This command displays a table of volume IDs and the corresponding number of extents that are used by each volume.

Note: If the MDisk specified is in a data reduction pool, the output includes all thin-provisioned and compressed volumes in the pool without displaying the number of extents in each.

An invocation example

```
lsvdiskextent -delim : vdisk0
```

The resulting output

```
id:number_extents
0:0
```

lsvdiskfcmapcopies

Use the **lsvdiskfcmapcopies** command to display a list of all FlashCopy mappings with a target volume that contains a valid copy of the specified volume.

Syntax

```
►►── lsvdiskfcmapcopies ── ─┬──────────┬─ ─┬────────────────────────┬─ ─┬─ vdisk_name ─┬──────────►◄
                            └─ -nohdr ─┘   └─ -delim ── delimiter ──┘   └─ vdisk_id ───┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

vdisk_name | *vdisk_id*
(Required) Specifies the name or ID of the volume for which the FlashCopy mappings are displayed.

Description

This command returns a list of the FlashCopy mappings that have a target volume with a valid copy of the specified volume. The target volumes of these mappings can be considered as candidate source volumes for mappings to restore from.

The mappings that are returned are in the copying, idle_copied, or stopping state with 100% progress.

Note: Maps that are rc_controlled are not shown in the view when this command is specified.

An invocation example

```
lsvdiskfcmapcopies -delim : 0
```

The resulting output

```
id:name:status:progress:difference:start_time:target_vdisk_id:
target_vdisk_name:group_id:group_name
2:fcmap2:copying:80:10:060627083137:10:vdisk10::
5:fcmap5:idle_copied:100:20:060627073130:12:vdisk12:1:fccstgrp1
```

lsvdiskfcmappings

Use the **lsvdiskfcmappings** command to display a list of FlashCopy mappings to which the volume belongs. A volume can be part of up to 256 FlashCopy mappings.

Syntax

```
►►── lsvdiskfcmappings ──┬──────────┬──┬─────────────────────────┬──┬─ vdisk_name ─┬──────────────►◄
                         └─ -nohdr ─┘  └─ -delim ── delimiter ──┘  └─ vdisk_id ───┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim : on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

vdisk_name | *vdisk_id*
(Required) Specifies the name or ID of the volume for which a list of all FlashCopy mappings is required.

Description

The **lsvdiskfcmappings** command returns a list of all FlashCopy mappings that the volume is a member of. The list is returned in no particular order.

An invocation example

```
lsvdiskfcmappings -delim : vdisk2
```

The resulting output:

```
fc_id:fc_name
1:fcmap1
3:fcmap3
```

lsvdiskhostmap

Use the **lsvdiskhostmap** command to list the volumes to the host mapping. These hosts specify volumes that are mapped to them; the volume is visible to these hosts.

Syntax

```
►►── lsvdiskhostmap ── ─┬─────────┬─ ─┬─────────────────────┬─ ──┬─ vdisk_id ───┬──────────────►◄
                        └─ -nohdr ─┘  └─ -delim ── delimiter ─┘    └─ vdisk_name ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If no data exists to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim : on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

vdisk_id | *vdisk_name*
(Required) Specifies the ID or name of the volume. The clustered system displays a list of all the hosts to which this volume is mapped and the Small Computer System Interface (SCSI) ID by which the volume is mapped.

Description

This command displays a list of host IDs and names. These hosts specify a volume that is mapped to them; that is, the volume is visible to these hosts. The SCSI LUN ID is also displayed. The SCSI LUN ID is the ID by which the volume is recognized by the host.

Determining the host that a volume is mapped to: List the hosts that this volume is mapped to, by issuing the following command:

```
lsvdiskhostmap vdisk_id | vdisk_name
```

where *vdisk_id* | *vdisk_name* is the name or ID of the volume. A list is displayed. Look for the host name or ID to determine which host this volume is mapped to. If no data is displayed, the volume is not mapped to any hosts.

The command returns the following values:

id Specifies the ID of the volume in the output for **lsvdiskhostmap**.

name Specifies the name of the volume in the output for **lsvdiskhostmap**.

SCSI_id
Specifies the SCSI ID.

host_id
Specifies the ID of the host.

host_name
Specifies the name of the host.

vdisk_UID
Specifies the UID of the volume.

IO_group_id
Specifies the ID of the input/output (I/O) group in which the host volume mapping exists.

IO_group_name
Specifies the name of I/O group in which the host volume mapping exists.

mapping_type
Indicates the mapping type for a host system. Values are `private` or `shared`.

host_cluster_id
Indicates the unique ID for a host system.

host_cluster_name
Indicates the unique name for a host system.

An invocation example

```
lsvdiskhostmap 0
```

The resulting output:

```
id name     SCSI_id host_id host_name vdisk_UID IO_group_id IO_group_name hostcluster_id hostcluster_name
4  vdisk4   0       3       host3     UID4      0           iogrp0
6  priv_6   4       4       host4     UID6      0           iogrp0
8  shared_8 5       4       host4     UID8      0           iogrp0        0              hostcluster0
8  shared_8 5       5       host5     UID8      0           iogrp0        0              hostcluster0
```

lsvdisklba

Use the **lsvdisklba** command to list the volume and logical block address (LBA) for the specified storage pool LBA.

Syntax

```
►►— lsvdisklba — — -mdisklba — mdisklba — ┬───────────────────────┬─ ┬─────────────┬─ ──────────►
                                          └ -delim — delimiter ┘   └ - nohdr┘

►— -mdisk ┬─ mdisk_id ───┬──────────────────────────────────────────────────────────────────────►◄
          └ mdisk_name ┘
```

Parameters

-mdisklba *mdisklba*
(Required) Specifies the 64-bit hexadecimal LBA on the MDisk. The LBA must be specified in hex, with a 0x prefix.

-nohdr

(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If no data exists to be displayed, headings are not displayed.

-delim *delimiter*

(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (**:**) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-mdisk *mdisk_id* | *mdisk_name*

(Required) Specifies the MDisk name or ID.

Description

The **lsvdisklba** command returns the LBA of the volume that is associated with the MDisk LBA.

If applicable, the command also lists the range of LBAs on both the volume and MDisk that are mapped in the same extent, or for thin-provisioned disks, in the same grain.

Note: If **lsvdisklba** is run during a software upgrade, the command fails and an error message is displayed.

The vdisk_lba field provides the corresponding LBA on the virtual capacity for the input LBA. For compressed volume copies, it is blank and the system provides the ranges of virtual LBAs that are compressed into the input LBA.

Table 109 provides command output that depends on several variables.

Table 109. **lsvdisklba** *command output scenarios*

Field	Typical scenario	Quorum disk	Thin-provisioned metadata	Extent not allocated	Formatting extent	Extent allocated to thin-provisioned disk, LBA not used on thin-provisioned disk
copy_id	yes	no	yes	no	yes	yes
vdisk_id	yes	no	yes	no	yes	yes
vdisk_name	yes	no	yes	no	yes	yes
type	allocated	metadata	metadata	unallocated	formatting	unallocated
vdisk_lba	yes	no	no	no	no	no
vdisk_start	yes	no	no	no	no	no
vdisk_end	yes	no	no	no	no	no
mdisk_start	yes	yes	yes	yes	yes	yes
mdisk_end	yes	yes	yes	yes	yes	yes

An invocation example

```
lsvdisklba -mdisk 1 -mdisklba 0x100123
```

The resulting output:

```
vdisk_id vdisk_name copy_id type      vdisk_lba  vdisk_start vdisk_end  mdisk_start        mdisk_end
0        vdisk0     0       allocated 0x00000123 0x00000000  0x000FFFFF 0x0000000000100000 0x00000000001FFFFF
```

lsvdiskmember

Use the **lsvdiskmember** command to display a list of MDisks that are members of the specified volume.

Syntax

```
►►── lsvdiskmember ── ┬─────────────────────┬─ ┬──────────┬─ ┬─────────────────────────┬─ ──────►
                      └─ -copy ── copy_id ─┘   └─ -nohdr ─┘   └─ -delim ── delimiter ─┘

►─┬─ vdisk_id ───┬────────────────────────────────────────────────────────────────────────────►◄
  └─ vdisk_name ─┘
```

Parameters

-copy *copy_id*
(Optional) Displays a list of MDisks that are members of the specified volume copy.

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If no data exists to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter -delim : on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

vdisk_id | *vdisk_name*
(Required) Identifies the specific volume to query.

Description

This command displays a list of managed disks, which provide extents that make up the volume that is specified by the ID.

Every volume is constructed from one or more MDisks. At times, you might have to determine the relationship between the two objects. The following procedure determines the relationships.

If you use the **lsmdiskmember** command, the concise view displays a list of volumes. These volumes are the ones that are using extents on the managed disk that is specified by the ID. The list displays the members of the respective object and is independent of the state of the individual members. If they are in an offline state, they are still displayed.

To determine the relationship between volumes and MDisks, issue the following command:

```
lsvdiskmember vdisk_id | vdisk_name
```

Where *vdisk_id* | *vdisk_name* is the name or ID of the volume. It displays a list of IDs that correspond to the MDisks that make up the volume.

To determine the relationship between volumes and MDisks, and the number of extents that are provided by each MDisk, you must use the command-line interface. Issue the following command:

```
lsvdiskextent vdisk_id | vdisk_name
```

Where *vdisk_id* | *vdisk_name* is the name or ID of the volume. It displays a table of MDisk IDs and the corresponding number of extents that each MDisk provides as storage for the specified volume.

To determine the relationship between MDisks and volumes, issue the following command:

```
lsmdiskmember mdisk_id | mdisk_name
```

Where *mdisk_id* | *mdisk_name* is the name or ID of the MDisk. It displays a list of IDs that correspond to the volumes that are using this MDisk.

To determine the relationship between MDisks and volumes, and the number of extents that are used by each volume, you must use the command-line interface. For a specified MDisk, issue the following command:

```
lsmdiskextent mdisk_id | mdisk_name
```

Where *mdisk_id* | *mdisk_name* is the name or ID of the MDisk. It displays a table of volume IDs and the corresponding number of extents that are used by each volume.

Note: You cannot specify this command for a thin-provisioned or compressed volume or a volume copy that is in a data reduction storage pool.

An invocation example

```
lsvdiskmember 1
```

The resulting output:

```
id
2
```

lsvdiskprogress

Use the **lsvdiskprogress** command to track the progress during new volume formatting.

Syntax

```
►►── lsvdiskprogress ──┬──────────┬──┬───────────────────────┬──┬──────────────┬──────────►◄
                       └─ -nohdr ─┘  └─ -delim ── delimiter ─┘  ├─ vdisk_id ───┤
                                                                └─ vdisk_name ─┘
```

Parameters

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (**:**) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by a colon character.

vdisk_id | *vdisk_name*
(Optional) Specifies the volume ID or name. If you do not specify this parameter, the progress of all volumes currently being formatted is displayed.

Description

This command displays the progress of the format of a new volume as a completed percentage. If the volume has multiple copies, the command reports the average progress of the format.

The command returns values for the following volume attributes:

id Indicates the ID of the volume that is being formatted.

progress
Indicates the formatting progress.

estimated_completion_time
Indicates the estimated time in which the formatting operation completes. The value is in the `YYMMDDHHMMSS` format, and is blank if the duration is not known.

An invocation example

```
lsvdiskprogress -delim : 0
```

The resulting output:

```
id:0
progress:58
estimated_completion_time:150101010100
```

lsvdisksyncprogress

Use the **lsvdisksyncprogress** command to display the progress of volume copy synchronization.

Syntax

```
►►─lsvdisksyncprogress─ ─┬──────────┬─ ─┬───────────────────┬─ ─┬────────────┬──────►
                         └─ -nohdr ─┘   └─ -delim ─ delimiter ─┘   └─ -copy ─ id ─┘

►─┬────────────┬──────────────────────────────────────────────────────────────────►◄
  ├─vdisk_name─┤
  └─vdisk_id───┘
```

Parameters

-copy *id*
(Optional) Specifies the volume copy ID to list synchronization progress. You must also specify a *vdisk_name* | *vdisk_id* value. If you do not specify this parameter, progress is displayed for all copies.

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (**:**) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

vdisk_name | *vdisk_id*
(Optional) Specifies the volume name or ID to list synchronization progress.

Description

To display the volume copies that require synchronization, specify the command with no parameters. To display the synchronization progress for all copies of a volume, specify the command with the *vdisk_name* | *vdisk_id* parameter. Estimated completion time is displayed in the `YYMMDDHHMMSS` format. The command displays progress for the following special cases as:

- A synchronized copy displays a progress of 100 and a blank estimated completion time.
- An offline copy or a copy with a zero synchronization rate displays a blank estimated completion time. An offline copy displays (gradually) decreasing progress if the volume is being written to.
- Nonmirrored volumes are displayed as a single copy with a progress of 100, and a blank estimated completion time.

The **lsvdisksyncprogress** command also displays the progress of a mirrored volume synchronization. After you create a mirrored volume by using the **mkvdisk** or **addvdiskcopy** command, you can use the command to monitor the progress of the synchronization.

An invocation example

```
lsvdisksyncprogress
```

The resulting output

```
vdisk_id vdisk_name copy_id progress estimated_completion_time
0        vdisk0     1       50       070301150000
3        vdisk3     0       72       070301132225
4        vdisk4     0       22       070301160000
8        vdisk8     1       33
```

An invocation example

```
lsvdisksyncprogress vdisk0
```

The resulting output

```
vdisk_id vdisk_name copy_id progress estimated_completion_time
0        vdisk0     0       100
0        vdisk0     1       50       070301150000
```

lsvolumebackup

Use the **lsvolumebackup** command to list the volumes that have cloud snapshot that enabled and volumes that have cloud snapshots in the cloud account.

Syntax

```
►►— lsvolumebackup — ┬─────────────────────────────────────┬─ ┬──────────┬─ ──────────────►
                     └─ -filtervalue — attribute_value ─┘   └─ -nohdr ─┘

►─┬─────────────────────┬─ ┬──────────────────┬──────────────────────────────────────────►◄
  └─ -delim — delimiter ─┘   └─ -filtervalue? ─┘
```

Parameters

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards:

- The wildcard character is the asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, enclose the filter entry within double quotation marks (""):
  `lsvolumebackup -filtervalue volume_id="1*"`

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-filtervalue?
(Optional) Displays a list of valid filter attributes. The following filters for the **lsvolumebackup** command are valid:

- `volume_UID`
- `volume_id`
- `volume_name`
- `volume_group_id`
- `volume_group_name`
- `cloud_account_id`
- `cloud_account_name`

Description

This command lists the volumes that use cloud snapshot and also lists volumes that have cloud snapshots in the cloud account.

This view spans all cloud accounts. To refresh the view by reloading what is on the cloud, specify `chcloudaccountawss3 -refresh 0` or `chcloudaccountswift -refresh 0`.

A snapshot that is being copied to the cloud (which means the snapshot status value is `copying` or `copying_error`) does not count towards the generation count total for the volume. Not counting towards the generation means that it is not available for a restore, and if there is a local system failure, the generation no longer exists in the cloud.

The specified volume appears in the displayed output when the volume has cloud snapshot that is enabled. The generation count is initially 0 and remains 0 while the volume copy is in progress. The generation count changes to 1 after the copy completes.

The last snapshot time is blank while the first snapshot is in progress. A snapshot that is being deleted from the cloud counts toward the generation count for the volume even if it is not available for a restore.

This table provides the attribute values that can be displayed as output view data.

Table 110. lsvolumebackup output

Attribute	**Description**
volume_UID	Indicates the volume UID.
volume_id	Indicates the volume ID if a volume with the specified UID exists on the local system. The value must be a number (or blank).
volume_name	Indicates the volume name. The value must be an alphanumeric string.
volume_group_id	Indicates the volume group ID that the volume is a member of (if applicable), if a volume with the specified UID exists on the local system. The value must be a number (or blank).
volume_group_name	Indicates the volume group name that the volume is a member of (if applicable), if a volume with the specified UID exists on the local system. The value must be an alphanumeric string (or blank).
cloud_account_id	Indicates the ID for the cloud account that contains the volume snapshots.
cloud_account_name	Indicates the cloud account name for the cloud account that contains the volume snapshot. The value must be an alphanumeric string.
last_backup_time	Indicates the timestamp of the most recent snapshot for this volume. The value must be in `YYMMDDHHMMSS` format or blank.
generation_count	Indicates the number of snapshot generations that exist for the specified volume. The value must be a number. **Note:** • Any generations that are being copied to the cloud do not count towards this number. • Any generations that are being deleted count towards this number until the delete process completes.
backup_size	Indicates the approximate amount of storage (the capacity) that is in use by snapshot generations for the specified volume

An invocation example

```
lsvolumebackup
```

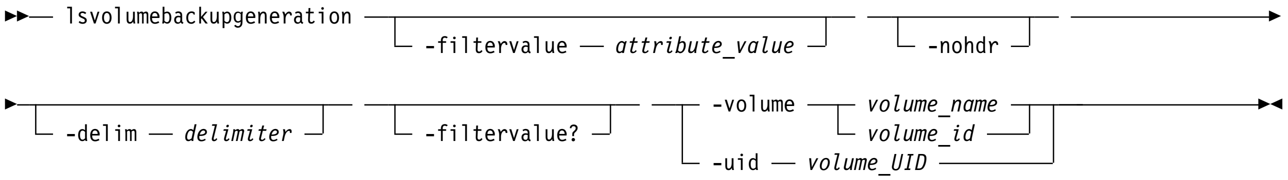
The resulting output:

```
volume_UID                       volume_id volume_name volume_group_id volume_group_name cloud_account_id cloud_account_nam
600507680CA880DF1800000000000002 2         vdisk2      2               logArchive        0                myAmazon
600507680CA880DF1800000000000003 3         vdisk3                                        0                myAmazon
600507680CA880DF1800000000000004 4         vdisk4                                        0                myAmazon
600507680CA880DF1800000000000017                                                         0                myAmazon
```

lsvolumebackupgeneration

Use the **lsvolumebackupgeneration** command to list any volume snapshots available on the specified volume.

Syntax



Parameters

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards:

- The wildcard character is the asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, enclose the filter entry within double quotation marks ("").

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (`:`) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-filtervalue?
(Optional) Displays a list of valid filter attributes. The following filters for the **lsvolumebackupgeneration** command are valid:

- `state`

-volume *volume_name* | *volume_id*
(Required) Specifies the volume to list cloud snapshots for by volume ID or name.

Note: The volume must exist on the local system.
The value for volume ID must be a number and the value for volume name must be an alphanumeric string. This parameter is mutually exclusive with **-uid**.

-uid *volume_UID*
(Optional) Specifies the volume to list cloud snapshots for by volume UID. This parameter is mutually exclusive with **-volume**.

Description

This command lists any available volume snapshots for the specified volume.

This table provides the attribute values that can be displayed as output view data.

Table 111. lsvolumebackupgeneration output

Attribute	Description
generation_id	Indicates the snapshot generation volume ID. The value must be a number.
backup_time	Indicates the timestamp of the most recent snapshot. The value must be in YYMMDDHHMMSS format (or blank).
volume_group_name	Indicate the volume group name. The value must be an alphanumeric string (or blank).
volume_size	Indicates the virtual capacity of the volume during snapshot generation. This value can differ from the current volume size (the capacity in MB or GB) if a volume with the specified UID exists on the local system.
type	Indicates the type of volume snapshot generation. The values are full and incremental.
state	Indicates the state of the volume backup generation (in the cloud system). The values are: • copying • complete • deleting
cloud_upload_size	Indicates the amount of data (the capacity in MB or GB) that is uploaded from the snapshot generation volume to the cloud system.

Generations that are being copied to the cloud account are included in the copying state value. Generations that are being deleted from the cloud account are also included in the deleting state value.

An invocation example

```
lsvolumebackupgeneration -volume 2
```

The resulting output:

```
generation_id backup_time   volume_group_name volume_size type        state    cloud_upload_size
1             160217021250                    50.00GB     full        complete 2.83GB
2             160217021355                    50.00GB     incremental complete 177.50MB
3             160218021402                    50.00GB     incremental complete 132.02MB
4             160219021400                    50.00GB     incremental copying  12.43MB
```

An invocation example

```
lsvolumebackupgeneration -uid 600507680CA880AB1200000000000015
```

The resulting output:

```
generation_id backup_time   volume_group_name volume_size type state    cloud_upload_size
1             160215021355                    10.00GB     full complete 53.88MB
```

lsvolumebackupprogress

Use the **lsvolumebackupprogress** command to display information about the progress of snapshot operations.

Syntax

```
►►─ lsvolumebackupprogress ─ ┬─────────────────────────────────────┬─ ┬──────────┬─ ───────────►
                             └─ -filtervalue ─ attribute_value ─┘   └─ -nohdr ─┘

►─┬────────────────────────┬─ ┬───────────────────┬──────────────────────────────────────────►◄
  └─ -delim ─ delimiter ─┘   └─ -filtervalue? ─┘
```

Parameters

-filtervalue *attribute=value*
(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards:

- The wildcard character is the asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, enclose the filter entry within double quotation marks (""):
  `lsvolumebackupprogress -filtervalue volume_id="1*"`

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-filtervalue?
(Optional) Displays a list of valid filter attributes. The following filters for the **lsvolumebackupprogress** command are valid:

- `volume_UID`
- `volume_id`
- `volume_name`
- `task`
- `status`

Description

This command display information about the progress of snapshot operations.

This table provides the attribute values that can be displayed as output view data.

Table 112. lsvolumebackupprogress output

Attribute	Description
volume_UID	Indicates the volume UID. The value must be a number 0 - 32 characters long.
volume_id	Indicates the volume ID (if a volume with the specified UID exists on the local system). The value must be a number (or blank).
volume_name	Indicates the volume name (if a volume with the specified UID exists on the local system). The value must be an alphanumeric string (or blank).
task	Indicates the type of task that is in progress. The values are `backup` and `delete`.
status	Indicates the task status. The values are: • `copying` • `copying_error` • `deleting` • `deleting_error`
generation_id	Indicates the generation ID for the volume that is created or deleted. The value must be a number.
backup_time	Indicates the snapshot time for the volume that is copied to the cloud system. The value must be in `YYMMDDHHMMSS` format for snapshot tasks or blank for deletion tasks.
progress	Indicates the task progress as a percentage. The value must be a number 0 - 99.
error_sequence_number	Indicates a particular error number. The value must be a number (or blank).

An invocation example

```
lsvolumebackupprogress
```

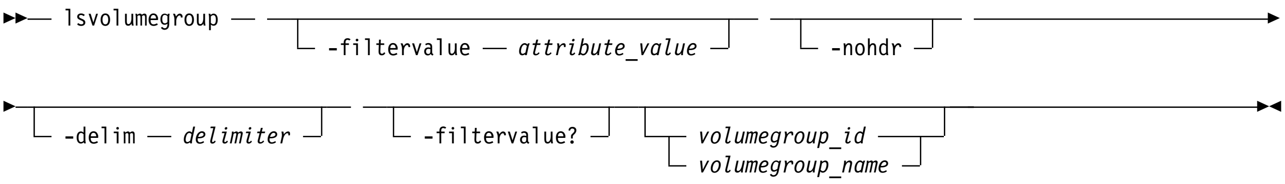
The resulting output:

```
volume_UID                       volume_id volume_name task   status        generation_id backup_time  progress   error_se
600507680CA880DF1800000000000002 2         vdisk2      backup copying       6             160218191005 88
600507680CA880DF1800000000000015 15        vdisk15     backup copying_error 19            160218190845 12         122
600507680CA880DF1800000000000015 15        vdisk15     delete deleting      8                          5
600507680CA880DF1800000000000017           vdisk108    delete deleting      10                         17
600507680CA880DF1800000000000018           vdisk109    delete deleting      many                       55
```

lsvolumegroup

Use the **`lsvolumegroup`** command to display information about configured volume groups.

Syntax



Parameters

-filtervalue *attribute=value*

(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards:

- The wildcard character is the asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, enclose the filter entry within double quotation marks (""):
  `lsvolumegroup -filtervalue id="1*"`

-delim *delimiter*

(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-nohdr

(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-filtervalue?

(Optional) Displays a list of valid filter attributes. The following filters for the **lsvolumegroup** command are valid:

- `id`
- `name`
- `volume_count`
- `backup_status`
- `last_backup_time`

volumegroup_id | *volumegroup_name*

(Optional) Specifies a volume group ID or volume group name. The value must be a number for the ID and an alphanumeric string for the name.

Description

This command displays information about configured volume groups.

This table provides the attribute values that can be displayed as output view data.

Table 113. lsvolumegroup output

Attribute	Description
id	Indicates the volume group ID. The value must be a number.
name	Indicates the volume group name. The value must be an alphanumeric string.
volume_count	Indicates the number of volume members in a group. The value must be a number.

Table 113. lsvolumegroup output (continued)

Attribute	**Description**
backup_status	Indicates whether a new (volume) group snapshot can be started. The values are: • `empty` • `off` • `not_ready` • `ready` • `copying` • `copying_error` If a group snapshot is in progress, the value represents the snapshot operation status.
last_backup_time	Indicates the most recent volume group snapshot time. The value must be in the `YYMMDDHHMMSS` format (or blank).

A concise invocation example

```
lsvolumegroup
```

The resulting output:

```
id                  name      volume_count        backup_status     last_backup_time
0                   VG1       5                   copying           160308115216
1                   VG2       0                   not_ready         150408115216
```

A detailed invocation example

```
lsvolumegroup 1
```

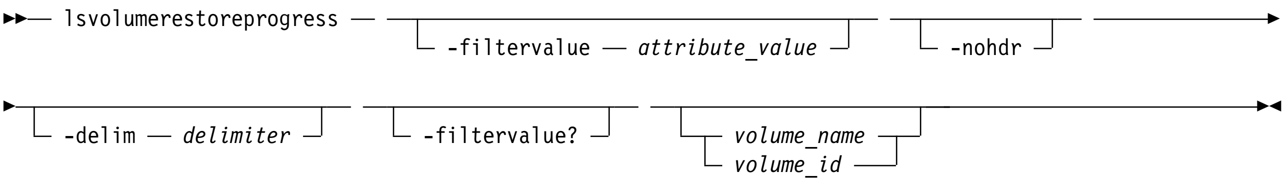
The resulting output:

```
id 1
name VG2
volume_count 0
backup_status not_ready
last_backup_time
```

lsvolumerestoreprogress

Use the **lsvolumerestoreprogress** command to display information about restore operation progress.

Syntax



Parameters

-filtervalue *attribute=value*

(Optional) Specifies a list of one or more filters. Only objects with a value that matches the filter attribute value are displayed.

Note: Some filters allow the use of a wildcard when you enter the command. The following rules apply to the use of wildcards:

- The wildcard character is the asterisk (*).
- The command can contain a maximum of one wildcard.
- When you use a wildcard, enclose the filter entry within double quotation marks (""):
  `lsvolumerestoreprogress -filtervalue volume_id="1*"`

-nohdr
(Optional) By default, headings are displayed for each column of data in a concise style view, and for each item of data in a detailed style view. The **-nohdr** parameter suppresses the display of these headings.

Note: If there is no data to be displayed, headings are not displayed.

-delim *delimiter*
(Optional) By default in a concise view, all columns of data are space-separated. The width of each column is set to the maximum width of each item of data. In a detailed view, each item of data has its own row, and if the headers are displayed, the data is separated from the header by a space. The **-delim** parameter overrides this behavior. Valid input for the **-delim** parameter is a 1-byte character. If you enter `-delim :` on the command line, the colon character (:) separates all items of data in a concise view; for example, the spacing of columns does not occur. In a detailed view, the data is separated from its header by the specified delimiter.

-filtervalue?
(Optional) Displays a list of valid filter attributes. The following filters for the **lsvolumerestoreprogress** command are valid:

- volume_id
- volume_name
- task
- status

volume_name | *volume_id*
(Optional) Indicates the volume name or ID for the volume that is restored. The value for volume name must be an alphanumeric string and the value for volume ID must be a number.

Description

This command displays information about restore operation progress.

This table provides the attribute values that can be displayed as output view data.

Table 114. lsvolumerestoreprogress output

Attribute	Description
volume_id	Indicates the volume ID for the volume that is restored. The value must be a number (or blank).
volume_name	Indicates the volume name for the volume that is restored. The value must be an alphanumeric string (or blank).
task	Indicates the type of task that is in progress. The value is `restore`.
status	Indicates the task status. The values are: • `restoring` • `restoring_error`
generation_id	Indicates the generation ID for the volume snapshot generation that is restored. The value must be a number.
backup_time	Indicates the time for the volume snapshot generation that is being restored to the cloud system (or available on the restore volume). The value must be in `YYMMDDHHMMSS` format for snapshot tasks or blank for deletion tasks.

Table 114. lsvolumerestoreprogress output (continued)

Attribute	Description
progress	Indicates the task progress as a percentage. The value must be a number 0 - 100. The status is blank when the status is `uncommitted`.
error_sequence_number	Indicates a particular error number. The value must be a number (or blank).
volume_backup_id	Indicates the UID of the volume snapshot that is restored. The value must be a number 1 - 32.
restore_volume_id	Indicates the ID of the volume that is the target of the restore operation. This volume is either the production volume (which means that the restore volume ID is the same as the volume ID) or a temporarily restored volume that is automatically provisioned by the restore process. The value must be a number.
restore_volume_name	Indicates the name of the volume that is the target of the restore operation. This volume is either the production volume (which means that the restore volume name is the same as the volume name) or a temporarily restored volume that is automatically provisioned by the restore process. The value must be an alphanumeric string.

A concise invocation example

```
lsvolumerestoreprogress
```

The resulting output:

```
volume_id volume_name task    status          generation_id backup_time  progress error_sequence_number
2         vdisk2      restore restoring       17            160102104511 88
21        vdisk21     restore restoring_error 4             160102105023 19                            122
```

A detailed invocation example

```
lsvolumerestoreprogress vdisk2
```

The resulting output:

```
volume_id 2
volume_name vdisk2
task restore
status restoring
generation_id 17
backup_time 160102104511
progress 88
error_sequence_number
volume_backup_UID 600507680CA880DF1800000000000002
restore_volume_id 2
restore_volume_name vdisk2
```

mkmetadatavdisk

Use the **mkmetadatavdisk** command to create one metadata volume (owner type is `host_integration_metadata`) from a storage pool. You can also export one block device or file system (that is based on this volume) in the configuration node.

Syntax

```
►►── mkmetadatavdisk ── ── -mdiskgrp ──┬─mdiskgrp_id───┬──────────────────────────────────────────►◄
                                       └─mdiskgrp_name─┘
```

Parameters

-mdiskgrp *mdiskgrp_id* | *mdiskgrp_name*
(Required) Assigns one or multiple storage pools for use in creating a metadata volume. The value must be a numeric value for *mdiskgrp_id* and an alphanumeric string for *mdiskgrp_name*.

Description

This command creates one metadata volume from a storage pool.

Note: You cannot specify a data reduction pool with this command.

An invocation example

```
mkmetadatavdisk -mdiskgrp pool_a
```

The resulting output:

```
Virtual Disk, id [2], successfully created
```

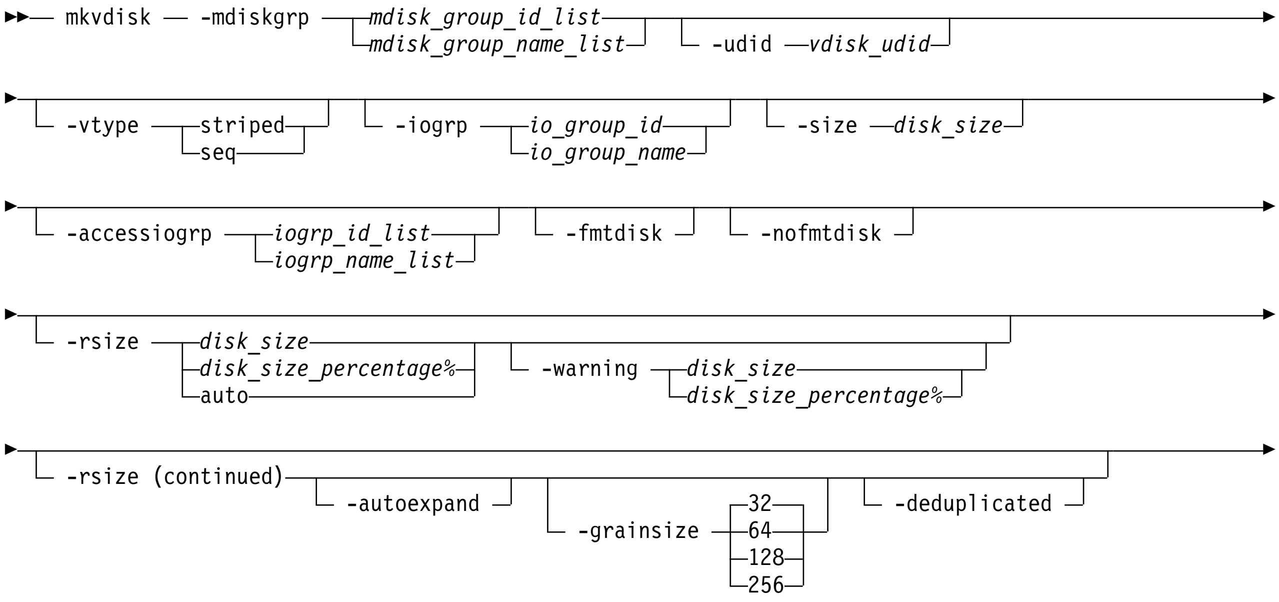
mkvdisk

Use the **mkvdisk** command to create sequential, striped, or image mode volume objects. When they are mapped to a host object, these objects are seen as disk drives with which the host can run I/O operations. Note the first syntax diagram below is for striped or sequential volumes and the second syntax diagram is for image mode volumes. Use the **mkvolume** command for a simplified way of creating high availability volumes. It includes `stretched` and `hyperswap` topologies. Use the **mkimagevolume** command for a simplified way of creating an image mode volume, importing existing data from a managed disk.

Note: The first syntax diagram depicts the creation of a **sequential** or **striped** mode volume. The second syntax diagram depicts the creation of an **image** mode volume.

Syntax

Create a sequential or striped mode volume.



```
►─┬──────────────┬──┬───────────────────────────────────────────────────────────┬──────►
  └─ -compressed ─┘  └─ -copies ──num_copies──┬───────────────┬──┬─────────────────────┬┘
                                              └─ -createsync ─┘  └─ -syncrate ──syncrate─┘

►─┬───────────────────────────────────────────┬──┬────────────────────────────────┬────►
  └─ -mirrorwritepriority ──┬─ latency ────┬──┘  └─ -mdisk ──┬─mdisk_id_list───┬──┘
                            └─ redundancy ─┘                 └─mdisk_name_list─┘

►─┬─────────────────────────┬──┬──────────────────┬──┬───────────────────────────┬─────►
  └─ -node ──┬─node_name─┬──┘  │                  │  └─ -name ──new_name_arg─────┘
             └─node_id───┘     └─ -unit ──┬─mb─┬──┘
                                          ├─b──┤
                                          ├─kb─┤
                                          ├─gb─┤
                                          ├─tb─┤
                                          └─pb─┘

►─┬─────────────────────────────┬──┬──────────────────────────────────┬──┬─────────────────────────┬──►◄
  │                             │  └─ -tier ──┬─ tier0_flash ──────┬──┘  └─ -easytier ──┬─ on ──┬──┘
  └─ -cache ──┬─readwrite─┬─────┘             ├─ tier1_flash ──────┤                    └─ off ─┘
              ├─readonly──┤                   ├─ tier_enterprise ──┤
              └─none──────┘                   └─ tier_nearline ────┘
```

Create an image mode volume.

```
►►── mkvdisk ── -mdiskgrp ──┬─mdisk_group_id───┬── -vtype ──image── -mdisk ──┬─mdisk_id_list───┬──────►
                            └─mdisk_group_name─┘                            └─mdisk_name_list─┘

►─┬──────────────────────────────────┬──┬─────────────────────┬──┬───────────────────────────────────────┬──►
  └─ -iogrp ──┬─io_group_id───┬──────┘  └─ -size ──disk_size──┘  └─ -accessiogrp ──┬─iogrp_id_list───┬──┘
              └─io_group_name─┘                                                   └─iogrp_name_list─┘

►─┬────────────┬──┬──────────────┬──────────────────────────────────────────────────────────────────►
  └─ -fmtdisk ─┘  └─ -nofmtdisk ─┘

►─┬─────────────────────────────────────────────────────────────────────────────────────────┬──────►
  └─ -rsize ──┬─disk_size───────────────┬──┬─────────────────────────────────────────────┬──┘
              ├─disk_size_percentage%───┤  └─ -warning ──┬─disk_size───────────────┬──┘
              └─auto────────────────────┘                └─disk_size_percentage%───┘

►─┬────────────────────────────────────────────────────────────────────────────────────────────┬──►
  └─ -rsize (continued)──┬───────────────┬──┬───────────────────────┬──┬──────────────────┬──┘
                         └─ -autoexpand ─┘  │                       │  └─ -deduplicated ─┘
                                            └─ -grainsize ──┬─32──┬──┘
                                                            ├─64──┤
                                                            ├─128─┤
                                                            └─256─┘

►─┬───────────┬──┬─────────────────────────────────────────────────────────────┬───────────────►
  └─ -import ─┘  └─ -copies ──num_copies──┬───────────────┬──┬──────────────────────┬──┘
                                          └─ -createsync ─┘  └─ -syncrate ──syncrate─┘

►─┬───────────────────────────────────────────┬──┬───────────────────────┬─────────────────────►
  └─ -mirrorwritepriority ──┬─ latency ────┬──┘  └─ -udid ──vdisk_udid───┘
                            └─ redundancy ─┘

►─┬─────────────────────────┬──┬──────────────────┬──┬───────────────────────────┬─────────────►
  └─ -node ──┬─node_name─┬──┘  │                  │  └─ -name ──new_name_arg─────┘
             └─node_id───┘     └─ -unit ──┬─mb─┬──┘
                                          ├─b──┤
                                          ├─kb─┤
                                          ├─gb─┤
                                          ├─tb─┤
                                          └─pb─┘
```

```
►──┬─────────────────────────────┬──┬─────────────────────────┬──────────────────────────────────────►◄
   │            ┌─readwrite─┐    │  └─ -easytier ──┬─ on ──┬─┘
   └─ -cache ───┼─readonly──┼────┘                 └─ off ─┘
                └─none──────┘
```

Parameters

-mdiskgrp *mdisk_group_id_list* | *mdisk_group_name_list*
(Required) Specifies one or more storage pools to use when you are creating this volume. If you are creating multiple copies, you must specify one storage pool per copy. The primary copy is allocated from the first storage pool in the list.

-udid *vdisk_udid*
(Optional) Specifies the unit number (*udid* for the disk. The `udid` is an identifier that is required to support OpenVMS hosts; no other systems use this parameter. Valid options are a decimal number 0 - 32 767, or a hexadecimal number 0 - 0x7FFF. A hexadecimal number must be preceded by `0x` (for example, `0x1234`).

-vtype seq | striped | image
(Optional) Specifies the virtualization type. When you create sequential or image mode volumes, you must also specify the **-mdisk** parameter. You cannot use `-vtype` *seq* or `-vtype` *image* in a data reduction pool. The default virtualization type is striped.

-iogrp *io_group_id* | *io_group_name*
(Optional) Specifies the I/O group (node pair) with which to associate this volume. If you specify **-node**, you must also specify **-iogrp**.

Remember:

- Create the first compressed volume copy for an I/O group to activate compression.
- You cannot create or move a volume copy that is compressed to an I/O group that contains at least one node that does not support compressed volumes. You must select another I/O group to move the volume copy to (but it does not affect moving to the recovery I/O group).

-size *disk_size*
(Required for sequential or striped volume creation) (Optional for image volume creation) Specifies the capacity of the volume, which is used with the value of the unit. All capacities, including changes, must be in multiples of 512 bytes. An error occurs if you specify a capacity that is not a multiple of 512. It can only happen when byte units (`-b`) are used. However, an entire extent is reserved even if it is only partially used. The default capacity is in MB. You can specify a capacity of 0. Specify the size in bytes in multiples of logical block address (LBA) sizes.

Note: If you do not specify the **-size** parameter when you create an image mode disk, the entire MDisk capacity is used.

-accessiogrp *iogroup_id_list* | *iogroup_name_list*
(Optional) Specifies the members of the volume I/O group access set. If this option is not specified, only the caching I/O group is added to the volume I/O group access set. If any access I/O groups are specified, only those I/O groups are in the access set (including if that set does not include the caching I/O group).

-fmtdisk
(Optional) Specifies that the volume be formatted. This parameter is no longer required for any volumes.

This parameter is not required when you create fully allocated volumes. The format operation is automatically applied to fully allocated volumes unless you specify **-nofmtdisk** parameter. The format operation sets the extents that make up this volume to all zeros after it is created. This process takes place in the background concurrently with host I/O operations on the new volume.

Remember: Formatting is on by default for single copy, fully allocated, and non-image mode volumes. You cannot format an image mode volume.

The format operation completes asynchronously. You can query the status by using the **lsvdiskprogress** command. You cannot specify this parameter with the **-vtype image** parameter.

This parameter is not required when you create thin-provisioned volumes. Thin-provisioned volumes return zeros for extents that are not written to. No format operation is required. This parameter also synchronizes mirrored copies by default.

-nofmtdisk
(Optional) Specifies that formatting be turned off for the new volume.

Remember: Formatting is on by default for single copy, fully allocated, and non-image mode volumes, and you can specify this parameter to turn it off.

-rsize *disk_size* | *disk_size_percentage%* | **auto**
(Optional) Defines how much physical space is initially allocated to the thin-provisioned or compressed volume. This parameter makes the volume thin-provisioned; otherwise, the volume is fully allocated. Specify the *disk_size* | *disk_size_percentage* value by using an integer, or an integer immediately followed by the percent character (%). Specify the units for a *disk_size* integer by using the **-unit** parameter; the default is MB. The **-rsize** value can be greater than, equal to, or less than the size of the volume. The **auto** option creates a volume copy that uses the entire size of the MDisk.

If you specify the **-rsize auto** option, you must also specify the **-vtype image** option. If you specify **-import** you must specify **-rsize**.

If the volume is in a data reduction storage pool, the value of the **-rsize** parameter will be ignored in **mkvdisk**. Only its presence or absence is used to determine whether the disk is a data reduction volume copy or a thick volume copy.

-warning *disk_size* | *disk_size_percentage%*
(Optional) Requires that the **-rsize** parameter also be specified. Specifies a threshold at which a warning error log is generated for volume copies. A warning is generated when the used disk capacity on the thin-provisioned copy first exceeds the specified threshold.

Note: You cannot specify this parameter for thin-provisioned or compressed volumes that are in data reduction pools.

You can specify a *disk_size* integer, which defaults to MBs unless the **-unit** parameter is specified. Or you can specify a *disk_size%*, which is a percentage of the volume size.

Important: If **-autoexpand** is:

1. Enabled, the default value for **-warning** is 80% of the volume capacity.
2. Not enabled, the default value for **-warning** is 80% of the real capacity.

To disable warnings, specify `0`.

-autoexpand
(Optional) Specifies that thin-provisioned copies automatically expand their real capacities by allocating new extents from their storage pool. Requires that the **-rsize** parameter also be specified. If the **-autoexpand** parameter is specified, the **-rsize** parameter specifies a capacity that is reserved by the copy. It protects the copy from going offline when its storage pool runs out of space by having the storage pool to consume this reserved space first.

The parameter has no immediate effect on image mode copies. However, if the image mode copy is later migrated to managed mode, the copy is then automatically expanded.

-grainsize 32 | **64** | **128** | **256**
(Optional) Sets the grain size (KB) for a thin-provisioned volume. This parameter also requires that the **-rsize** parameter be specified. If you are using the thin-provisioned volume in a FlashCopy map,

use the same grain size as the map grain size for best performance. If you are using the thin-provisioned volume directly with a host system, use a small grain size. The grain size value must be 32, 64, 128, or 256 KB. The default is 256 KB.

If the volume to be created is a thin-provisioned volume in a data reduction storage pool, the **-grainsize** parameter cannot be used. This type of volume will be created with a grain size of 8 KB.

-deduplicated
(Optional) Creates a deduplicated volume. If you specify **-deduplicated**, you must also specify **-rsize** because it applies only to thin-provisioned or compressed volumes.

Note: Data deduplication works only with data reduction storage pools. You can only create deduplicated volumes and volume copies in an I/O group if there are no compressed volumes or volume copies in regular storage pools.

-compressed
(Optional) Enables compression for the volume. This parameter must be specified with **-rsize** and cannot be specified with **-grainsize**.

-import
(Optional) Imports a thin-provisioned volume from the MDisk. If you specify **-import** you must also specify **-rsize**.

-copies *num_copies*
(Optional) Specifies the number of copies to create. The *num_copies* value can be 1 or 2. Setting the value to 2 creates a mirrored volume. The default value is 1.

-createsync
(Optional) Creates copies in sync. Use this parameter if you have already formatted the MDisks, or when read stability to unwritten areas of the volume is not required.

-syncrate *syncrate*
(Optional) Specifies the copy synchronization rate. A value of zero (0) prevents synchronization. The default value is 50. See Table 116 on page 737 for the supported **-syncrate** values and their corresponding rates. Use this parameter to alter the rate at which the fully allocated volume or mirrored volume format before synchronization.

-mirrorwritepriority *latency* | *redundancy*
(Optional) Specifies how to configure the mirror write algorithm priority. If not specified, the default value is *latency*.

1. Choosing *latency* means a copy that is slow to respond to a write input/output (I/O) becomes unsynchronized, and the write I/O completes if the other copy successfully writes the data.
2. Choosing *redundancy* means a copy that is slow to respond to a write I/O synchronizes completion of the write I/O with the completion of the slower I/O to maintain synchronization.

-mdisk *mdisk_id_list* | *mdisk_name_list*
(Optional) Specifies one or more managed disks. For sequential and image mode volumes, the number of MDisks must match the number of copies. For sequential mode volumes, each MDisk must belong to the specified storage pool. For striped volumes, you cannot specify the **-mdisk** parameter if the **-copies** value is greater than 1.

When you create a single copy striped volume, you can specify a list of MDisks to stripe across.

You must use this parameter to specify an MDisk that has a mode of unmanaged.

-node *node_id* | *node_name*
(Optional) Specifies the preferred node ID or the name for I/O operations to this volume. You can use the **-node** parameter to specify the preferred access node. If you specify **-node**, you must also specify **-iogrp**.

Note: This parameter is evaluated by multipath device drivers. The system chooses a default if you do not supply this parameter.

-unit b | kb | mb | gb | tb | pb
(Optional) Specifies the data units to use along with the capacity that is specified by the **-size** and **-rsize** parameters. The default unit type is MB.

-name *new_name_arg*
(Optional) Specifies a name to assign to the new volume.

-cache readwrite | readonly | none
(Optional) Specifies the caching options for the volume. Valid entries are:

- *readwrite* enables the cache for the volume.
- *readonly* disables write caching while allowing read caching for a volume.
- *none* disables the cache mode for the volume.

The default is *readwrite*.

-tier *tier0_flash* | *tier1_flash* | *tier_enterprise* | *tier_nearline*
(Optional) Specifies the MDisk tier when an image mode copy is added.

tier0_flash
Specifies a `tier0_flash` hard disk drive or an external MDisk for the newly discovered or external volume.

tier1_flash
Specifies an `tier1_flash` (or flash drive) hard disk drive or an external MDisk for the newly discovered or external volume.

tier_enterprise
Specifies a `tier_enterprise` hard disk drive or an external MDisk for the newly discovered or external volume.

tier_nearline
Specifies a `tier_nearline` hard disk drive or an external MDisk for the newly discovered or external volume.

ssd Specifies an SSD (or flash drive) hard disk drive or an external MDisk for the newly discovered or external volume.

nearline
Specifies a nearline hard disk drive or an external MDisk for the newly discovered or external volume.

enterprise
Specifies an enterprise hard disk drive or an external MDisk for the newly discovered or external volume.

Note: This action applies to both copies if you are creating a mirrored volume with two image mode copies by using this command.

-easytier *on* | *off*
Determines whether the IBM Easy Tier function is allowed to move extents for this volume.

Note: The **-easytier** parameter must be followed by either *on* or *off*:

- If set to *on*, then Easy Tier functions are active.
- If set to *off*, then Easy Tier functions are inactive.

If the Easy Tier feature is enabled, and if a volume copy is striped and not being migrated, the following table applies.

Table 115. Easy Tier settings for storage pools and volumes

Storage pool Easy Tier setting	**Number of tiers in the storage pool**	**Volume copy Easy Tier setting**	**Volume copy Easy Tier status**
Off	One	Off	inactive (see note 1)
Off	One	On	inactive (see note 1)
Off	Two	Off	inactive (see note 1)
Off	Two	On	inactive (see note 1)
Measure	One	Off	measured (see note 2)
Measure	One	On	measured (see note 2)
Measure	Two	Off	measured (see note 2)
Measure	Two	On	measured (see note 2)
Auto	One	Off	measured (see note 2)
Auto	One	On	balanced (see note 3)
Auto	Two	Off	measured (see note 2)
Auto	Two	On	active (see note 4)
On	One	Off	measured (see note 2)
On	One	On	balanced (see note 3)
On	Two	Off	measured (see note 2)
On	Two	On	active (see note 4)

Notes:

1. When the volume copy status is `inactive`, no Easy Tier functions are enabled for that volume copy.
2. When the volume copy status is `measured`, the Easy Tier function collects usage statistics for the volume but automatic data placement is not active.
3. When the volume copy status is `balanced`, the Easy Tier function enables performance-based pool balancing for that volume copy.
4. When the volume copy status is `active`, the Easy Tier function operates in automatic data placement mode for that volume.

If the volume copy is in image or sequential mode or is being migrated, the volume copy Easy Tier status is `measured` instead of `active`.

The default Easy Tier setting for a storage pool is `auto`, and the default Easy Tier setting for a volume copy is `on`. It means that Easy Tier functions except pool performance balancing are disabled for storage pools with a single tier, and that automatic data placement mode is enabled for all striped volume copies in a storage pool with two or more tiers.

Description

This command creates a new volume object. You can use the command to create various types of volume objects, making it one of the most complex commands.

Remember: You can create a striped volume only in a child pool - you cannot create sequential or image volumes in a child pool.

You must decide which storage pool or storage pools provide the storage for the volume. Use the **lsmdiskgrp** command to list the available storage pools and the amount of free storage in each pool. If you are creating a volume with more than one copy, each storage pool that you specify must have enough space for the size of the volume.

If you create a thin-provisioned or compressed volume from a data reduction storage pool, that volume uses the same properties as the data reduction storage pool. You can create fully allocated volumes from data reduction pools, but these volumes use different data reduction properties.

A thin-provisioned or compressed volume that is in a data reduction storage pool must:

- Not be in sequential or image mode.
- Not have a warning threshold set (using **-warning**).
- Use `-cache readwrite` when caching.
- Have **-autoexpand** enabled.

For thin-provisioned and compressed volume copies that are in data reduction storage pools, the Easy Tier status is taken from the data reduction pool because that data is managed by a central data disk. It means that you cannot specify **-easytier** with thin-provisioned or compressed volumes.

Important: The extent size for the storage pool can limit volume size. Consider the maximum volume size that you want to use when you create storage pools. Refer to the information on creating storage pools for a comparison of the maximum volume capacity for each extent size. The maximum is different for thin-provisioned volumes.

A compressed volume in a data reduction pool can only be created in an I/O group with V5030, V7000, or SVC node types. Thin provisioned volumes can be created on any node type.

No restriction exists for the number of compressed volumes within a data reduction storage pool.

Choose an I/O group for the volume. This action determines which nodes in the system process the I/O requests from the host systems. If you have more than one I/O group, ensure that you distribute the volumes between the I/O groups so that the I/O workload is shared evenly between all nodes. Use the **lsiogrp** command to show the I/O groups and the number of volumes that are assigned to each I/O group.

Note: It is normal for systems with more than one I/O group to have storage pools that have volumes in different I/O groups. FlashCopy processing can make copies of volumes whether the source and target volumes are in the same I/O group. However, if you plan to use intra-system Metro or Global Mirror operations, ensure that both the master and auxiliary volume are in the same I/O group.
The command returns the IDs of the newly created volume.

An encryption key cannot be used when you create an image mode MDisk. To use encryption (when the MDisk has an encryption key), the MDisk must be self-encrypting.

Specify the virtualization type by using the **-vtype** parameter; the supported types are sequential (*seq*), *striped*, and *image*.

sequential (seq)

This virtualization type creates the volume that uses sequential extents from the specified MDisk (or MDisks, if creating multiple copies). The command fails if there are not enough sequential extents on the specified MDisk.

striped

The default virtualization type. If the **-vtype** parameter is not specified, *striped* is the default; all managed disks in the storage pool are used to create the volume. The striping is at an extent level; one extent from each managed disk in the group is used. For example, a storage pool with 10 managed disks uses one extent from each managed disk. It then uses the 11th extent from the first managed disk, and so on.

If the **-mdisk** parameter is also specified, you can supply a list of managed disks to use as the stripe set. It can be two or more managed disks from the same storage pool. The same circular algorithm is used across the striped set. However, a single managed disk can be specified more than once in the list. For example, if you enter `-mdisk 0:1:2:1`, the extents are from the following managed disks: 0, 1, 2, 1, 0, 1, 2, and so forth. All MDisks that are specified in the **-mdisk** parameter must be in the managed mode.

A capacity of 0 is allowed.

image

This virtualization type allows image mode volumes to be created when a managed disk already has data on it, perhaps from a previrtualized subsystem. When an image mode volume is created, it directly corresponds to the (previously unmanaged) managed disk that it was created from. Therefore, except for thin-provisioned image mode volumes, volume logical block address (LBA) *x* equals managed disk LBA *x*. You can use this command to bring a nonvirtualized disk under the control of the system. After it is under the control of the system, you can migrate the volume from the single managed disk. When it is migrated, the volume is no longer an image mode volume.

You can add image mode volumes to an already populated storage pool with other types of volumes, such as a striped or sequential.

Important: An image mode volume must be 512 bytes or greater. At least one extent is allocated to an image mode volume.

Remember: If you create a mirrored volume from two image mode MDisks without specifying a **-size** value, the capacity of the resulting volume is the smaller of the two MDisks, and the remaining space on the larger MDisk is not accessible.

Attention:

1. Do not create a volume in an offline I/O group. You must ensure that the I/O group is online before you create a volume to avoid any data loss. This action applies in particular to re-creating volumes that are assigned the same object ID.
2. To create an image mode disk, you must already have a quorum disk present in the system because an image mode disk cannot be used to hold quorum data. Refer to information on quorum disk creation for more details.
3. The command fails if either limit of 2048 volumes per I/O Group or 8192 volume copies per system is reached.

The rate at which the volume copies resynchronize after loss of synchronization can be specified by using the **-syncrate** parameter. This table provides the relationship of the *syncrate* value to the data copied per second.

Note: These settings also affect the initial rate of formatting.

Table 116. Relationship between the syncrate value and the data copied per second

User-specified *syncrate* attribute value	Data copied/sec
1 - 10	128 KB
11 - 20	256 KB
21 - 30	512 KB
31 - 40	1 MB
41 - 50	2 MB
51 - 60	4 MB
61 - 70	8 MB
71 - 80	16 MB
81 - 90	32 MB
91 - 100	64 MB

An invocation example

```
mkvdisk -mdiskgrp Group0 -size 0
-iogrp 0 -vtype striped -mdisk mdisk1 -node 1
```

The resulting output:

```
Virtual Disk, id [1], successfully created
```

An invocation example for creating an image mode volume

```
mkvdisk -mdiskgrp Group0
-iogrp 0 -vtype image -mdisk mdisk2 -node 1
```

The resulting output:

```
Virtual Disk, id [2], successfully created
```

An invocation example for creating a new volume

```
mkvdisk -mdiskgrp Group0 -size 0 -unit kb
-iogrp 0 -vtype striped -mdisk mdisk1 -node 1 -udid 1234 -easytier off
```

The resulting output:

```
Virtual Disk id [2], successfully created
```

An invocation example for creating a thin-provisioned volume

```
mkvdisk -mdiskgrp Group0 -iogrp 0 -vtype striped -size 10 -unit gb -rsize 20% -autoexpand -grainsize 32
```

The resulting output:

```
Virtual Disk id [1], successfully created
```

An invocation example for creating a compressed volume copy

```
mkvdisk -mdiskgrp 0 -iogrp 0 -size 1 -unit tb -rsize 0 -autoexpand -warning 0 -compressed
```

The resulting output:

```
Virtual Disk id [1], successfully created
```

An invocation example for creating a mirrored image-mode volume

```
mkvdisk -mdiskgrp Group0:Group0 -mdisk mdisk2:mdisk3 -iogrp 0 -vtype image -copies 2
```

The resulting output:

```
Virtual Disk id [1], successfully created
```

An invocation example for creating a mirrored volume

```
mkvdisk -iogrp 0 -mdiskgrp 0:1 -size 500 -copies 2
```

The resulting output:

```
Virtual Disk id [5], successfully created
```

An invocation example for configuring a mirror write algorithm priority

```
mkvdisk -mdiskgrp Group0 -iogrp 0 -vtype striped -mirrorwritepriority redundancy -size 500
```

The resulting output:

```
Virtual Disk id [5], successfully created
```

An invocation example to create a disk with default grain size

```
mkvdisk -iogrp 0 -mdiskgrp 0 -size 100 -rsize 5%
```

The resulting output:

```
Virtual Disk id [5], successfully created
```

An invocation example for creating a volume with I/O groups 0 and 1 in its I/O group access set

```
mkvdisk -iogrp 0 -mdiskgrp 0 -size 500 -accessiogrp 0:1
```

The resulting output:

```
Virtual Disk id [5], successfully created
```

An invocation example for creating a volume with warning considerations

```
mkvdisk -mdiskgrp 6 -size 200 -rsize 50 -iogrp 0
```

The resulting output:

```
Virtual Disk, id [2], successfully created
...
lsvdisk 2
...
warning 20   # threshold in MB = 50 x 80 / 100 = 40 MB; threshold as %age of volume capacity = 40 / 200 * 100 = 20
...
```

An invocation example for creating a volume with warning considerations

```
mkvdisk -mdiskgrp 6 -size 200 -rsize 50 -iogrp 0  -warning 80%
```

The resulting output:

```
Virtual Disk, id [2], successfully created
...
lsvdisk 2
...
warning 80   # displayed as %age of volume capacity
...
```

An invocation example for creating a volume with warning considerations

```
mkvdisk -mdiskgrp 6 -size 200 -rsize 50 -iogrp 0 -autoexpand
```

The resulting output:

```
Virtual Disk, id [2], successfully created
...
lsvdisk 2
...
warning 80   # displayed as %age of volume capacity
...
```

An invocation example to create a volume with the read cache enabled

```
mkvdisk -iogrp 0 -size 10 -unit gb -mdiskgrp 0 -cache readonly
```

The resulting output:

```
Virtual Disk, id [2], successfully created
```

An invocation example to create volume `Group0`

```
mkvdisk -mdiskgrp Group0 -iogrp io_grp0 -vtype image -mdisk 13 -node 1 -udid 1234 -tier tier_nearline
```

The resulting output:

```
Virtual Disk, id [0], successfully created
```

An invocation example to turn off formatting while creating volume `Burnley1`

```
mkvdisk -mdiskgrp Burnley1 -iogrp 0 -mdiskgrp 0:1 -size 500 -nofmtdisk -copies 2
```

The resulting output:

```
Virtual Disk, id [0], successfully created
```

An invocation example to create a deduplicated volume copy

```
mkvdisk -mdiskgrp datareductionpool0 -size 100 -unit gb -iogrp 0 -rsize 0 -autoexpand -deduplicated
```

The resulting output:

```
Virtual Disk, id [4], successfully created
```

mkvdiskhostmap

Use the **mkvdiskhostmap** command to create a new mapping between a volume and a host, which makes the volume accessible for input/output (I/O) operations to the specified host.

Syntax

```
►►── mkvdiskhostmap ── ┬────────────┬ ── -host ──┬─ host_id ───┬── ┬─────────────────────────┬──────►
                       └─ -force ───┘            └─ host_name ─┘    └─ -scsi ── scsi_num_arg ─┘

►── ┬─ vdisk_name ─┬─────────────────────────────────────────────────────────────────────────────►◄
    └─ vdisk_id ───┘
```

Parameters

-force
(Optional) Allows multiple volume-to-host assignments, which are not normally allowed.

-host *host_id* | *host_name*
(Required) Specifies the host to map the volume to, either by ID or by name.

-scsi *scsi_num_arg*
(Optional) Specifies the Small Computer System Interface (SCSI) logical unit number (LUN) ID to assign to this volume on the given host. The *scsi_num_arg* parameter contains the SCSI LUN ID that is assigned to the volume on the given host for all I/O groups that provide access to the volume. You

must check your host system for the next available SCSI LUN ID on the given host bus adapter (HBA). If you do not specify the **-scsi** parameter, the next available SCSI LUN ID in each I/O group that provides access is provided to the host.

vdisk_name | *vdisk_id*
(Required) Specifies the name of the volume that you want to map to the host, either by ID or by name.

Description

This command creates a new mapping between the volume and the specified host. The volume is presented to the host as if the disk is directly attached to the host. It is only after this command is processed, that the host can perform I/O transactions to the volume.

Optionally, you can assign a SCSI LUN ID to the mapping. When the HBA in the host scans for devices that are attached to it, it discovers all volumes that are mapped to its Fibre Channel ports. When the devices are found, each one is allocated an identifier (SCSI LUN ID). For example, the first disk found is usually `SCSI LUN 0`, and so on. You can control the order in which the HBA discovers volumes by assigning the SCSI LUN ID, as required. If you do not specify a SCSI LUN ID, the cluster automatically assigns the next available SCSI LUN ID, if any mappings already exist with that host. When you issue the **mkvdiskhostmap** command, the assigned SCSI LUN ID number is returned.

The **mkvdiskhostmap** command fails if the:

- Host to which this mapping is being made is not associated with any one of the I/O groups in the volume access set
- Volume has more than one I/O group in its access set and the host being mapped to the volume does not support volumes being mapped from multiple I/O groups

Remember: iSCSI hosts can access volumes that are accessible through multiple I/O groups (as well as single I/O groups).

If you generate different SCSI LUN IDs, only one is returned. The returned ID is for the highest-numbered I/O group to which the volume was mapped. To view other values, issue **lshostvdiskmap** or **lsvdiskhostmap**.

The SCSI LUN ID is used for the highest numbered I/O group to which the volume is mapped.

Some HBA device drivers stop when they find a gap in the SCSI LUN IDs. For example:

- Volume 1 is mapped to Host 1 with SCSI LUN ID 1
- Volume 2 is mapped to Host 1 with SCSI LUN ID 2
- Volume 3 is mapped to Host 1 with SCSI LUN ID 4

When the device driver scans the HBA, it must stop after identifying volumes 1 and 2, because no SCSI LUN is mapped with ID 3. For optimal performance, ensure that the SCSI LUN ID allocation is contiguous.

You can create multiple volume assignments (assigning the same volume to multiple hosts, for example, which might be particularly useful for clustered system hosts assigning a volume to multiple hosts). Normally, multiple volume-to-host assignments are not used because corruption is likely to occur if more than one host can access a disk. However, in certain multiple path environments, a volume must be mapped to more than one host. This includes the IBM SAN File System. To map to more than one host, you must use the **mkvdiskhostmap** command with the **-force** parameter. For example:

```
mkvdiskhostmap -host host1 -force 4
mkvdiskhostmap -host host2 -force 4
```

Note: When assigning the same volume to multiple hosts, you should use the same SCSI ID for all hosts.

These commands create two host-to-volume mappings for volume `4` that map to `host1` and `host2`. Omitting the **`-force`** parameter causes the mapping to fail if that volume is already mapped to a host.

The command also fails if the host object (to which this mapping is being made) is not associated with the I/O group containing the volume.

If a new mapping is created between a host (type `hide_secondary`) and a volume that is a secondary volume in a remote copy relationship, the:

- Mapping is created for configuration purposes (it can be changed or deleted)
- Secondary volume is not presented to the host

The mapped volume is presented to the host if the:

- Host type is changed to a type other than `hide_secondary`
- Remote copy relationship is stopped by specifying **`-access`**
- Volume is no longer a secondary volume because the remote copy relationship is deleted or switched

Note: You cannot specify this command if the volume is an auxiliary volume in an `active-active` relationship or if a volume is a change volume in any type of relationship.

An invocation example

```
mkvdiskhostmap -host host1 -scsi 1 5
```

The resulting output:

```
Virtual Disk to Host map, id [1], successfully created
```

mkvolume

Use the **`mkvolume`** command to create an empty volume from existing storage pools. You can use this command for high availability configurations that include HyperSwap or `stretched` systems, but it can also be used for volumes that are not high availability.

Syntax

```
►►— mkvolume — ┬─────────────────┬— — -size — disk_size — ┬───────────────────┬— ──────────►
               └ name — name ────┘                         └ -unit ─┬─ b ──┬──┘
                                                                    ├─ kb ─┤
                                                                    ├─ mb ─┤
                                                                    ├─ gb ─┤
                                                                    ├─ tb ─┤
                                                                    └─ pb ─┘

►┬──────────────────────────────┬— — -pool ─┬─ storage_pool_id ───┬— — ──────────────►
 └ -iogrp ─┬─ iogroup_id ───┬───┘           └─ storage_pool_name ─┘
           └─ iogroup_name ─┘

►┬─────────────────────────┬ ┬──────────────────────────────────────────┬─────────────►
 └ -cache ─┬─ none ─────┬──┘ └┬─ -thin ────────┬─┬──────────────────┬───┘
           ├─ readonly ─┤     └─ -compressed ──┘ └─ -deduplicated ──┘
           └─ readwrite ┘
```

```
►─┬──────────────────────────────────────────┬─ ─┬─────────────────────────────────────────────┬─ ─ ──────►
  └─ -buffersize ──┬─ buffer_size ─────────┬─┘   └─ -warning ──┬─ warning_capacity ─────────┬─┘
                   └─ buffer_percentage% ──┘                   └─ warning_percentage% ──────┘

►─┬────────────────────┬─ ─┬─────────────────────────┬─ ─┬──────────────────┬──────────────────►
  └─ -noautoexpand ───┘   └─ -grainsize ──┬─ 32 ──┬─┘   └─ -udid ── udid ─┘
                                          ├─ 64 ──┤
                                          ├─ 128 ─┤
                                          └─ 256 ─┘

►─ -volumegroup ──┬─ volumegroup_name ─┬────────────────────────────────────────────────────►◄
                  └─ volumegroup_id ───┘
```

Parameters

-name *name*

(Optional) Specifies the name that is used for the volume that is created. This value must be an alphanumeric string 1 - 63 characters.

Remember: If you do not specify **-name**, a unique default name such as volume1 is used.

-size *disk_size*

(Required) Specifies the capacity of the volume, which is used with the value of the unit. The default capacity is in MB. When the unit of bytes is used, all capacities must be in multiples of 512 bytes. An entire extent is reserved even if it is only partially used.

-unit *b* | *kb* | *mb* | *gb* | *tb* | *pb*

(Optional) Specifies the data units to use with the capacity that is specified by the **-size** parameter. The default unit type is *mb*.

-iogrp *iogroup_id_list* | *iogroup_name_list*

(Optional) Specifies the I/O group that the new volume is cached in. The value can be a colon-separated list of up to two I/O group IDs or names. If no value is specified, the caching I/O group is selected based on the storage pool site. If you do not specify the **-iogrp** parameter, the caching I/O group is selected by the system.

Important: If two I/O groups are specified, they must be in different sites, and the storage pools that are specified must be in different sites. The order of the sites must correspond.

If you create a HyperSwap volume, the caching I/O groups are selected based on the sites of the storage pools.

-pool *storage_pool_id_list* | *storage_pool_name_list*

(Required) Specifies the storage pool in which to create the new volume. The value must be a colon-separated list of up to two storage pool IDs or names.

Note: If one storage pool is specified, a basic volume is created with one copy.

On systems with standard topology, a mirrored volume can be created by specifying two storage pools.

On systems with a stretched or hyperswap topology, a highly available volume can be created by specifying two storage pools in different sites.

-cache *none* | *readonly* | *readwrite*

(Optional) Specifies the caching options for the volume. Use one of the following valid entries:

- readwrite enables the cache for the volume (default).
- readonly disables write caching but allows read caching for a volume.
- none disables the cache mode for the volume.

-thin
(Optional) Specifies that the volume is to be created with thin provisioning. You cannot specify this parameter with **-compressed**. If you do not specify **-thin** or **-compressed**, the volume that is created is fully allocated.

-compressed
(Optional) Specifies that the volume is to be created compressed. If the -iogrp parameter is not specified, the least used I/O group is used for compressed copies (considering the subset of I/O groups that support compression).

Remember: This command fails if no I/O groups support compression. If two sites exist, both sites must have at least one I/O group that supports compression.

You cannot specify this parameter with **-thin**. If you do not specify **-thin** or **-compressed**, the volume that is created is fully allocated.

-deduplicated
(Optional) Creates a deduplicated volume. If you specify **-deduplicated**, you must also specify **-rsize** because it applies only to thin-provisioned or compressed volumes.

Note: Data deduplication works only with data reduction storage pools. You can only create deduplicated volumes and volume copies in an I/O group if there are no compressed volumes or volume copies in regular storage pools.

-buffersize *buffer_size* | *buffer_percentage*
(Optional) Specifies the pool capacity the volume attempts to reserve as a buffer for thin-provisioned and compressed volumes. You must specify either **-thin** or **-compressed** with this parameter. The default value is 2%.

Note: You cannot specify a buffer size for thin-provisioned or compressed volumes that are in data reduction pools.

-warning *warning_capacity* | *warning_percentage*
(Optional) Specifies a threshold at which a warning error log is generated for volumes. A warning is generated when the used disk capacity on the thin-provisioned volume exceeds the specified threshold. You must specify either **-thin** or **-compressed** with this parameter. The default value is 80%.

-noautoexpand
(Optional) Specifies that the volume not automatically expand as it is written to. The available buffer capacity decreases as the used capacity increases. The volume copy goes offline if the buffer capacity is fully used. The buffer capacity can be increased by specifying expandvdisksize -rsize. You must specify either **-thin** or **-compressed** with this parameter. If you do not specify **-noautoexpand**, the volume automatically expands as it is written to.

-grainsize *32* | *64* | *128* | *256*
(Optional) Sets the grain size (KB) for a thin-provisioned volume. If you are using the thin-provisioned volume in a FlashCopy map, use the same grain size as the map grain size for best performance. If you are using the thin-provisioned volume directly with a host system, use a small grain size. The grain size value must be 32, 64, 128, or 256 KB. The default is 256 KB.

-udid *udid*
(Optional) Specifies the unit number udid for the volume.

Important: The udid is an identifier that is required to support OpenVMS hosts (no other systems use this parameter).
Valid options are a decimal number from 0 through 32767, or a hexadecimal number from 0 through 0x7FFF. A hexadecimal number must be preceded by 0x (for example, 0x1234).

-volumegroup *volumegroup_name* | *volumegroup_id*
(Optional) Specifies the volume group that a volume belongs to. The value must be an alphanumeric string for the volume group name and the value must be a number for the volume group ID.

Description

This command creates an empty volume, which is a formatted (zeroed) volume, by using storage from existing storage pools. You can also create a highly available volume on systems with `stretched` or `hyperswap` topology.

If you create a thin-provisioned or compressed volume from a data reduction storage pool, the properties of the storage pool are used for the new volume. You can create fully allocated volumes from data reduction storage pools, but those volumes do not use the storage pool properties.

On some node types, you can create a compressed volume copy in a data reduction storage pool for an I/O group. A compressed volume copy in a data reduction pool can only be created in an I/O group with V5030, V7000, or SVC node types. You can create thin-provisioned volume copies on any node type. Volumes can also have fully allocated volume copies in data reduction storage pools.

You cannot specify **-noautoexpand** when you create thin-provisioned or compressed volume copies from a data reduction storage pool.

You cannot create a volume copy that is a thin-provisioned or compressed volume in a data reduction storage pool, and the volume caching mode is `none` or `readonly`. You must specify **chvdisk** to change the volume caching mode to `readwrite`.

You cannot specify **-warning** for a thin-provisioned or compressed volume copy in a data reduction storage pool.

You cannot specify **-grainsize** for thin-provisioned and compressed volume copies in data reduction storage pools. This type of volume copy is created with a size of 8 KB.

Thin-provisioned or compressed volume copies in data reduction pools cannot be created if the data reduction storage pool is offline and requires recovery. If the recovery is still in progress, you must wait until the recovery is complete and the pool is in `online` state.

Use the **mkimagevolume** command to create a new volume by importing existing data on a managed disk.

Scenario 1

If the I/O group contains:

- At least one 8 GB node.
- At least one thin-provisioned or compressed volume in a data reduction pool.
- And you try to set the FlashCopy bitmap size for that I/O group to at least 1.5 GB.

The command fails due to insufficient resources available.

Scenario 2

When a thin-provisioned or compressed volume is created within a data reduction pool, the pool must have enough capacity to create more volumes that track SCSI unmap operations from the host. If this capacity is not available, the command fails.

Scenario 3

Volumes cannot be created in a data reduction pool if offline thin-provisioned or compressed volumes exist in a data reduction pool, either because of thin provisioning (out of space or corruption), or a component underneath thin provisioning is holding a volume in the pool offline.

An invocation example to create a volume in storage pool `0`

```
mkvolume -pool 0 -size 1000
```

The detailed resulting output:

```
Volume, id [0], successfully created.
```

An invocation example for creating a thin-provisioned `stretched` volume on a system with `stretched` topology

```
mkvolume -pool site1pool:site2pool -size 1 -unit tb -thin
```

The detailed resulting output:

```
Volume, id [1], successfully created.
```

An invocation example to create a HyperSwap volume with a `hyperswap` topology

```
mkvolume -pool site1pool:site2pool -size 200
```

The detailed resulting output:

```
Volume, id [2], successfully created.
```

An invocation example to create a thin-provisioned volume from a data reduction storage pool

```
mkvolume -pool datareductionpool2 -size 10 -unit gb -thin
```

The detailed resulting output:

```
Volume, id [6], successfully created.
```

An invocation example to create a deduplicated volume copy

```
mkvolume -pool datareductionpool0 -size 100 -unit gb -iogrp 0 -thin -deduplicated
```

The resulting output:

```
Virtual Disk, id [4], successfully created
```

mkvolumegroup

Use the **`mkvolumegroup`** command to create and configure a new volume group.

Syntax

```
►►── mkvolumegroup ── ┬─────────────────────────────┬──────────────────────►◄
                      └─ -name ── volumegroup_name ─┘
```

Parameters

-name *volumegroup_name*
(Optional) Specifies a volume group name. The value must be an alphanumeric value. If you do not specify a volume group name, one is automatically created and assigned to the volume group.

Description

This command creates and configures a new volume group.

An invocation example

```
mkvolumegroup
```

The resulting output:

```
Volume Group, id [0], successfully created
```

An invocation example

```
mkvolumegroup -name Sunday
```

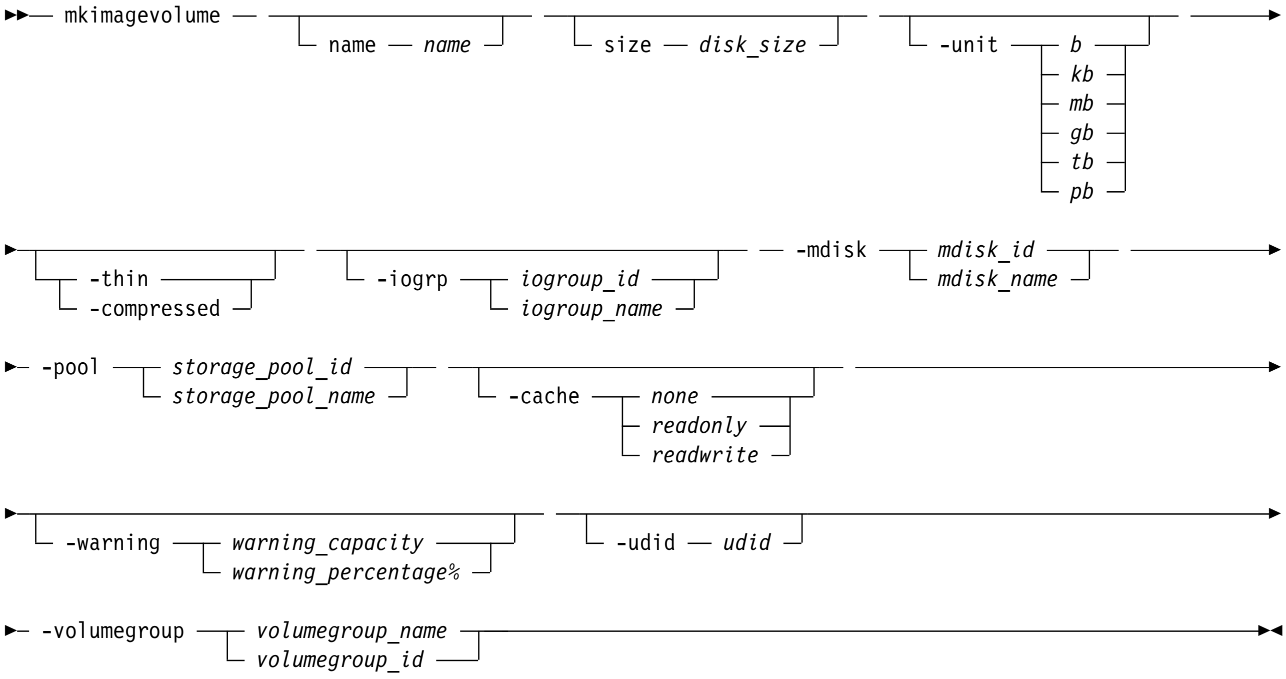
The resulting output:

```
Volume Group, id [1], successfully created
```

mkimagevolume

Use the **mkimagevolume** command to create an image mode volume by importing (preserving) data on a managed disk from another storage system.

Syntax



Parameters

-name *name*
: (Optional) Specifies the name that is used for the volume that is created. This value must be an alphanumeric string 1 - 63 characters.

  Remember: If you do not specify **-name**, a unique default name such as `volume1` is used.

-size *disk_size*
: (Required if **-thin** or **-compressed** is specified) Specifies the capacity of the volume, which is used

with the value of the unit. The default capacity is in MB. When the unit of bytes is used, all capacities must be in multiples of 512 bytes. An entire extent is reserved even if it is only partially used.

Remember: This parameter is optional if **-thin** or **-compressed** are not specified. If this parameter is not specified, the volume is fully allocated.
For thin and compressed volumes, the real capacity is set from the MDisk size.

-unit *b* | *kb* | *mb* | *gb* | *tb* | *pb*
(Optional) Specifies the data units to use with the capacity that is specified by the **-size** parameter. The default unit type is *mb*.

-thin
(Optional) Specifies that the volume is to be created with thin provisioning. You cannot specify this parameter with **-compressed**. If you do not specify **-thin** or **-compressed**, the volume that is created is fully allocated.

-compressed
(Optional) Specifies that the volume is to be created compressed. If the -iogrp parameter is not specified, the least used I/O group is used for compressed copies (considering the subset of I/O groups that support compression).

Remember: This command fails if no I/O groups support compression.

You cannot specify this parameter with **-thin**. If you do not specify **-thin** or **-compressed**, the volume that is created is fully allocated.

-iogrp *iogroup_id* | *iogroup_name*
(Optional) Specifies the I/O group that the new volume is cached in.

-mdisk *mdisk_id mdisk_name*
(Required) Specifies which currently unused MDisk to use to create the image mode volume.

-pool *storage_pool_id* | *storage_pool_name*
(Required) Specifies the storage pool in which to create the new volume. The value for *storage_pool_id* must be a numerical value.

-cache *none* | *readonly* | *readwrite*
(Optional) Specifies the caching options for the volume. Valid entries are:

- readwrite enables the cache for the volume (default)
- readonly disables write caching but allows read caching for a volume
- none disables the cache mode for the volume

-warning *warning_capacity* | *warning_percentage*
(Optional) Specifies a threshold at which a warning error log is generated for volume copies. A warning is generated when the used disk capacity on the thin-provisioned volume exceeds the specified threshold. You must specify either **-thin** or **-compressed** with this parameter. The default value is 80%.

-udid *udid*
(Optional) Specifies the unit number udid for the disk. The udid is an identifier that is required to support OpenVMS hosts; no other systems use this parameter. Valid options are a decimal number from 0 through 32767, or a hexadecimal number from 0 through 0x7FFF. A hexadecimal number must be preceded by 0x (for example, 0x1234).

Remember: When you create a HyperSwap volume, this value is set only on the master volume.

-volumegroup *volumegroup_name* | *volumegroup_id*
(Optional) Specifies the volume group that a volume image belongs to. The value must be an alphanumeric string for the volume group name and the value must be a number for the volume group ID.

Description

Use the **mkimagevolume** command to create a new image mode volume. This command is used to import a volume, preserving existing data.

Note: A volume copy in a data reduction pool cannot be an image mode volume copy.

Import a fully allocated image mode volume into storage pool 0 with MDisk 2 at full capacity

```
mkimagevolume -mdisk 2 -pool 0
```

The detailed resulting output:

```
Volume, id [0], successfully created.
```

Import a thin-provisioned image mode volume (with virtual capacity 25GB) into storage pool 1 with MDisk 7

```
mkimagevolume -mdisk 7 -pool 1 -thin -size 25 -unit gb
```

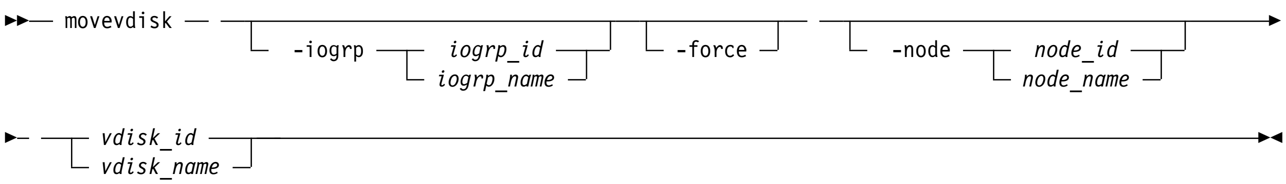
The detailed resulting output:

```
Volume, id [2], successfully created.
```

movevdisk

Use the **movevdisk** command to move the preferred node of a volume either within the same caching I/O group or to another caching I/O group.

Syntax



Parameters

-iogrp *iogrp_id* | *iogrp_name*
(Optional) Specifies the I/O group to move the volume to.

-force
(Optional) Use the force parameter to force the volume to be removed from an I/O group. This option overrides the cache flush mechanism.

Remember:

- If you specify the **-force** parameter, the contents of the cache are discarded and the volume might be corrupted by the loss of the cached data. Use the **-force** parameter with caution.
- If the force parameter is used to move a volume that has out-of-sync copies, a full resynchronization is required.

-node *node_id* | *node_name*
(Optional) Specifies the node ID or name that is assigned as the preferred node.

vdisk_id | *vdisk_name*
(Required) Specifies the volume to move.

Description

Use the **movevdisk** to migrate a single volume to a new I/O group - repeat this action for other volumes as required. This command can also move the preferred node of a volume without changing the caching I/O group, but it does not change which I/O groups can access the volume (only the caching I/O group is changed).

Important: You cannot migrate or move an image mode volume.

This command is not supported to change the I/O group if either copy is thin-provisioned or compressed and within a data reduction pool. The preferred node can be changed for volumes that are in data reduction pools.

A compressed volume can also be moved, and you can specify the preferred node in the new I/O group. You can move a volume that is in a FlashCopy mapping, but the FlashCopy bitmaps remain in the original I/O group. You cannot move volumes when the FlashCopy mapping is in `preparing` or `prepared` state. Additionally, a volume can be moved if it is the target of a FlashCopy mapping that is in `stopping` state.

You cannot move a volume to change the caching I/O group for a volume that is in a Global Mirror, Metro Mirror, or HyperSwap relationship, regardless of whether it is a primary, secondary, or change volume. To move a volume in a Global Mirror, Metro Mirror, or HyperSwap relationship, the relationship must first be deleted. You can change a preferred node without changing caching I/O group for this type of a volume.

If the volume is offline, use one of the **recovervdisk** commands to recover the volume and bring it back online. To specify a preferred node for the volume, use the **-node node_id | node_name** parameter with the **movevdisk** command. Use the **movevdisk** command to change the I/O group with which this volume is associated.

Important: Do not move:

- A volume to an offline I/O group under any circumstance;

  Remember: To avoid data loss, make sure that the I/O group is online before you move the volume.
- An offline volume to the recovery I/O group

You can migrate a volume to a new I/O group to manually balance the workload across the nodes in the clustered system. You might end up with a pair of nodes that are overworked and another pair of nodes that are underworked.

Remember: You cannot move a volume if that volume is being formatted.

If the volume is a target of a FlashCopy mapping with a source volume in an `active-active` relationship, then the new I/O group must be in the same site as the source volume. The system permits moving a volume in a remote copy relationship if the move does not change the I/O group (it changes the preferred node). If the volume is in an `active-active` relationship, the new I/O group must be in the same site as the source I/O group.

Note: Remote copy includes Metro Mirror, Global Mirror, and HyperSwap.

An invocation example to move `DB_Volume` to I/O group `2`

```
movevdisk -iogrp 2 DB_Volume
```

The resulting output:

```
No feedback
```

An invocation example to move `DB_Volume` to I/O group `IOGRP3` with a new preferred node ID 7

```
movevdisk -iogrp IOGRP3 -node 7 DB_Volume
```

The resulting output:

```
No feedback
```

An invocation example to change preferred node of volume DB_Volume with a new preferred node ID as 8 in the same IOGRP

```
movevdisk -node 8 DB_Volume
```

The resulting output:

```
No feedback
```

recovervdisk

Use the **recovervdisk** command to acknowledge volume data loss and brings the volume back online.

Syntax

```
►►—recovervdisk— ————————————————————————— ——┬vdisk_name┬————————————————————————————————————————►◄
                 └—-copy— —copy_id—┘          └vdisk_id——┘
```

Parameters

vdisk_name | *vdisk_id*
(Required) Specifies the volume to recover.

-copy *copy_id*
(Optional) Specifies the ID of the copy to recover.

Description

The specified volume, and all copies if mirrored, are recovered and brought back online. If the volume is thin-provisioned or has thin-provisioned copies, this command triggers the thin-provisioned repair process. If the volume is mirrored, the **recovervdisk** command triggers a resynchronization from a synchronized copy. The progress of the resynchronization can be monitored by using the **lsvdisksyncprogress** command. The volume remains online during the resynchronization process.

The **recovervdisk** command also starts the repair of any thin-provisioned copies that have a `fast_write_state` of corrupt. The progress of the repair process can be monitored by using the **lsrepairsevdiskcopyprogress** command.

A volume that is still offline because it is being repaired following the **recovervdisk** command has a `fast_write_state` of repairing. The volume is brought online when the repair process is complete.

An invocation example (to recover volume 45)

```
recovervdisk vdisk45
```

The following command is an invocation example to recover copy 0 of volume 45:

```
recovervdisk -copy 0 vdisk45
```

recovervdiskbycluster (Discontinued)

Attention: The **recovervdiskbycluster** command has been discontinued. Use the **recovervdiskbysystem** command instead.

recovervdiskbyiogrp

Use the **recovervdiskbyiogrp** command to acknowledge data loss for all volumes in the specified I/O group with a fast_write_state of corrupt and brings the volumes back online.

Syntax

```
►►──recovervdiskbyiogrp── ──┬─io_group_name─┬──────────────────────────────────────────────►◄
                            └─io_group_id───┘
```

Parameters

io_group_name | *io_group_id*
(Required) Specifies the I/O group for volume recovery.

Description

All volumes in the specified I/O group that have a fast_write_state of corrupt; and all copies, if mirrored, are recovered and brought back online. If any of the volumes are thin-provisioned or have thin-provisioned copies, the **recovervdiskbyiogrp** command triggers the thin-provisioned repair process. If volumes are mirrored, the command triggers a resynchronization from a synchronized copy. The progress of the resynchronization can be monitored by using the **lsrepairsevdiskcopyprogress** command. Volumes remain online during the resynchronization process.

If none of the volumes in the specified I/O group have a fast_write_state of corrupt, the **recovervdiskbyiogrp** command still starts the repair process for any corrupted copies of mirrored volumes. The progress of the repair process can be monitored by using the **lsrepairsevdiskcopyprogress** command. If no corrupted volumes exist or no repairs to copies are needed, no error is returned.

Volumes that are still offline because they are being repaired following the **recovervdiskbyiogrp** command have a fast_write_state of repairing. Volumes are brought online when the repair process is complete.

An invocation example

```
recovervdiskbyiogrp iogrp2
```

The following output is displayed:

```
No feedback
```

recovervdiskbysystem

Use the **recovervdiskbysystem** command to acknowledge data loss for all volumes in the clustered system (system) with a fast_write_state of corrupt and bring the volumes back online.

Syntax

```
►►──recovervdiskbysystem── ────────────────────────────────────────────────────────────────►◄
```

Parameters

There are no parameters for this command.

Description

All volumes in the system that have a `fast_write_state` of `corrupt`; and all copies, if mirrored, are recovered and brought back online. If any of the volumes are thin-provisioned or have thin-provisioned copies, the **`recovervdiskbysystem`** command triggers the thin-provisioned repair process. If volumes are mirrored, the command triggers a resynchronization from a synchronized copy. The progress of the resynchronization can be monitored by using the **`lsvdisksyncprogress`** command. Volumes remain online during the resynchronization process.

If none of the volumes in the system have a `fast_write_state` of `corrupt`, the **`recovervdiskbysystem`** command still starts the repair process for any corrupt copies of mirrored volumes. The progress of the repair process can be monitored using the **`lsrepairsevdiskcopyprogress`** command. If there are no corrupt volumes or no repairs to copies are required, no error is returned.

Volumes that are still offline because they are being repaired following the **`recovervdiskbysystem`** command have a `fast_write_state` of `repairing`. Volumes are brought online when the repair process is complete.

An invocation example

```
recovervdiskbysystem
```

The resulting output:

```
No feedback
```

repairsevdiskcopy

The **`repairsevdiskcopy`** command repairs the metadata on a thin-provisioned volume copy or a compressed volume copy.

Syntax

```
►►—repairsevdiskcopy— ──────────────────── ──┬vdisk_name┬─────────────────────────────►◄
                        └─ -copy ── 0 │ 1 ─┘    └vdisk_id──┘
```

Parameters

-copy 0 | 1
(Optional) Specifies the volume copy to repair.

vdisk_name | *vdisk_id*
(Required) Specifies the volume to repair.

Description

The **`repairsevdiskcopy`** command repairs the metadata on a thin-provisioned volume or compressed volume copy. Run this command only when you are directed by the fix procedures or by your product support information.

Running the command automatically detects corrupted metadata. The command holds the volume offline during the repair. During this time there are limited operations permissible.

If a repair operation completes successfully and the volume was previously offline because of corrupted metadata, the command brings the volume back online. The only limit on the number of concurrent repair operations is the number of volume copies in the configuration. Once started, a repair operation cannot be paused or canceled; the repair can only be ended by deleting the copy.

An invocation example

```
repairsevdiskcopy vdisk8
```

The resulting output:

```
No feedback
```

repairvdiskcopy

Use the **repairvdiskcopy** command to detect and (optionally) correct any volume copies that are not identical.

Syntax

```
►►──repairvdiskcopy── ──┬─-medium───┬── ──┬─────────────────────┬──┬─vdisk_name─┬──────────────────────►◄
                        ├─-resync───┤     └─ -startlba ── lba ──┘  └─vdisk_id───┘
                        └─-validate─┘
```

Parameters

-medium
(Optional) Converts sectors that contain different readable data into virtual medium errors on the specified volume. It fixes preexisting medium errors found on only one volume copy by replacing them with data from the other volume copy. This parameter cannot be used with the **-validate** and **-resync** parameters. You must specify one of the three parameters.

-resync
(Optional) Corrects sectors that contain different readable data by copying contents from the primary volume copy to other copies on the specified volume. It fixes preexisting medium errors found on only one volume by replacing them with data from the other volume. This parameter cannot be used with the **-medium** and **-validate** parameters. You must specify one of the three parameters.

-validate
(Optional) Reports the first difference in readable data found on synchronized online copies of the specified volume, on or after the specified **-startlba** value. This parameter cannot be used with the **-medium** and **-resync** parameters. You must enter one of the three parameters.

-startlba *lba*
(Optional) Specifies a starting logical block address (LBA) on which to begin the command. The LBA must be specified in hex, with a 0x prefix.

vdisk_name | *vdisk_id*
(Required) Specifies the volume to repair. You must specify this parameter last on the command line.

Description

The **repairvdiskcopy** command detects and optionally, corrects any volume copies that are not identical. For the purposes of comparison, preexisting medium errors found on only one volume are ignored and fixed by replacing them with data from the other volume copy. The results are logged to the SAN Volume Controller error log. The **-validate** parameter compares synchronized online copies of the specified volume. The **-medium** parameter changes any sectors that are not identical into virtual medium errors. The **-resync** parameter copies any sectors that are not identical to the other volume copies. You cannot use this command with a volume that is fast formatting.

You must specify **-validate**, **-medium**, or **-resync**.

Attention:

1. Before you run the **repairvdiskcopy** command, ensure that all volume copies are synchronized.
2. Only one **repairvdiskcopy** command can run on a volume at a time. You must wait for the **repairvdiskcopy** command to complete processing before running the command again.
3. Once you start the **repairvdiskcopy** command, you cannot use the command to stop processing.
4. The primary copy of a mirrored volume cannot be changed while the **repairvdiskcopy -resync** command is running.

Use the **-startlba** parameter to specify a starting Logical Block Address (LBA). Enter an LBA value from `0` to full disk size minus one. The parameter logs the first error found and then stops the command. By repeating this parameter, you can collect all of the instances where the volume copies are not identical.

During **repairvdiskcopy** command operation, the volume remains online. The I/O and synchronization operations are allowed while the command is in progress.

The rate for the **repairvdiskcopy** command is controlled by the synchronization rate of the volume that is being repaired. To suspend the repair process, set the synchronization rate of the volume to `0` using the **chvdisk** command.

An invocation example

```
repairvdiskcopy -resync -startlba 0x0 vdisk8
```

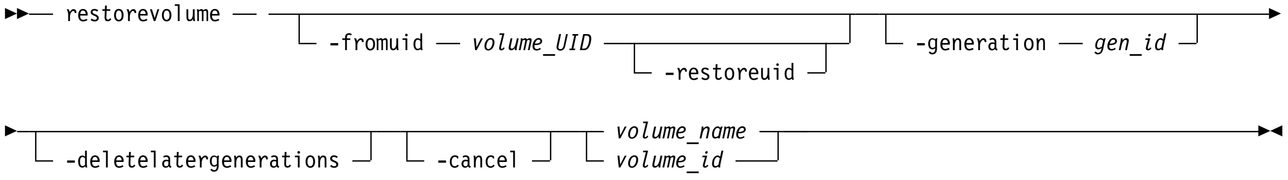
The resulting output:

```
No feedback
```

restorevolume

Use the **restorevolume** command to restore a volume from a snapshot generation.

Syntax



Parameters

-fromuid *volume_UID*
: (Optional) Specifies the volume snapshot to restore (specified by volume UID). The value must be a number.

Use this parameter to restore a snapshot from a different volume. This means that the specified UID must be different from the UID of the volume being restored (when you specify *volume_name* or *volume_id*).

Note: The volume being restored to cannot have cloud snapshot enabled if you specify this parameter.

-restoreuid
(Optional) Specifies the UID of the restored volume should be set to the UID of the volume snapshot that is being restored. You must specify **-fromuid** with this parameter.

-generation *gen_id*
(Optional) Specifies the snapshot generation to restore. The value must be a number.

-deletelatergenerations
(Optional) Specifies that all snapshot generations should be deleted after the generation is restored.

Note: This parameter is required when the volume has cloud snapshot enabled and the generation being restored is not the most recent snapshot of the volume.

-cancel
(Optional) Cancels the restore operation.

volume_name | *volume_id*
(Required) Specifies the volume name or ID to restore. The value for the volume ID must be a number and the value for the volume name must be an alphanumeric string.

Description

This command restores a volume from a snapshot generation.

The restore operation is performed directly on the volume specified by volume name or volume ID (without use of a temporary volume). The volume is `offline` while the restore operation is in progress. If the restore process is canceled before it completes the data on the volume is inconsistent and not usable.

An invocation example

To restore an earlier generation (generation 3) for volume `volume7`:

```
restorevolume -generation 3 -deletelatergenerations volume7
```

The resulting output:

```
No feedback
```

An invocation example

To restore the most recent snapshot (generation 5) for volume ID 7:

```
restorevolume -generation 5 volume7
```

The resulting output:

```
No feedback
```

An invocation example

```
restorevolume -generation 1 -fromuid 6005076400B70038E00000000000001C 1
```

The resulting output:

```
No feedback
```

rmvdisk

Use the **rmvdisk** command to delete a volume. This command cannot be used for high availability volumes. Use the **rmvolume** command for high availability volumes.

Syntax

```
►►─ rmvdisk ─ ─ ┬─────────────────────┬─ ┬─ vdisk_id ───┬─────────────────────►◄
                ├─ -removehostmappings ┤  └─ vdisk_name ─┘
                └─ -force ─────────────┘
```

Parameters

-force

(Optional) The specified volume is to be deleted, even if mappings still exist between this volume and one or more hosts. Host-to-volume mappings and any FlashCopy mappings that exist for this volume are deleted.

Important: If you stop a FlashCopy mapping that has dependent FlashCopy mappings, the dependent mapping target volumes might become unusable.

If you stop a FlashCopy mapping whose target volume is also in a Metro Mirror or Global Mirror relationship, the relationship stops. If a remote copy relationship that is associated with the target was mirroring I/O when the map was copying, the relationship might lose its difference recording capability and require a full resynchronization on a subsequent restart.

To determine your dependent FlashCopy mappings before you use the **-force** parameter, run the **lsfcmapdependentmaps** command.

Important: To prevent an active volume from being deleted unintentionally, you can use a global system setting to enable volume protection (see the **chsystem** command). You can specify a time period for which the volume must be idle before you can delete it. If volume protection is enabled and the time period is not expired, the volume deletion fails even if the **-force** parameter is used.
If the **-force** deletion of a volume causes dependent mappings to be stopped, any target volumes for those mappings that are in Metro Mirror or Global Mirror relationships are also stopped. The dependent mappings can be identified by using the **lsvdiskdependentmaps** command on the volume that you want to delete.

Note: Using the **-force** parameter might result in a data loss. Use it only under the direction of your product support information, or if you are willing to accept the risk of volume data loss.

If you do not specify this parameter, a volume cannot be deleted while a backup operation is in progress. Additionally, a volume that contains image mode copies cannot be deleted while a restore operation is in progress (if the volume contains inconsistent data).

-removehostmappings

(Optional) Removes all host mappings for the specified volume before the volume is deleted.

Note: Using the **-removehostmappings** parameter might result in a data loss. Use it only under the direction of your product support information, or if you are willing to accept the risk of volume data loss.

vdisk_id | *vdisk_name*

Specifies the name of the volume to delete, either by ID or by name.

Note: To deactivate compression, use the **rmvdiskcopy** to delete the last compressed volume copy for an I/O group.

Description

This command deletes an existing managed mode volume or an existing image mode volume. The extents that made up this volume are returned to the pool of free extents that are available on the storage pool, if the volume is in managed mode.

Remember: If you run this command, any data that was on the volume is lost. Before you run this command, ensure that the volume (and any data that resides on it) is no longer required.

This command is unsuccessful if:

- Volume protection is enabled (by using the **chsystem** command).
- The volume that is being removed received I/O within the defined volume protection time period.
- The data reduction pool is corrupted.

With an `active-active` relationship, either or both of the master and auxiliary volumes can provide the information for host systems to read through the master volume ID. To remove the auxiliary volume from the relationship, delete the relationship so hosts can access the master copy.

Remember: Any FlashCopy mappings with the specified volume as their source volume are deleted when you specify this command.

Deleting a managed mode volume

When you use this command to delete a managed mode volume, all the data on the volume is deleted. The extents that make up the volume are returned to the pool of free extents that are available in the storage pool.

If host mappings exist for the volume, or if any FlashCopy mappings would be affected, the deletion fails. You can use the **-force** parameter to force the deletion. If you use the **-force** parameter, mappings that have the volume as source or target are deleted, other mappings in a cascade might be stopped, and then the volume is deleted. The **-force** parameter also deletes any Metro Mirror or Global Mirror relationships that exist for the specified volume (and any information that is not staged in the fast write cache).

If the volume is in the process of migrating to an image mode volume (by using the **migratetoimage** command), the deletion fails unless you use the **-force** parameter. If you use the **-force** parameter, the migration is halted and then the volume is deleted. Before you run this command, ensure that the volume (and any data that resides on it) is no longer required.

Deleting an image mode volume

If the volume is mirrored and one or both copies is in image mode, you must first wait for all fast-write data to be moved to the controller logical unit. This pause ensures that the data on the controller is consistent with the data on the image mode volume before the volume is deleted. This process can take several minutes to complete, and is indicated by the fast_write_state of the volume, which is empty. If the **-force** parameter is specified, the fast-write data is discarded and the volume is deleted immediately; the data on the controller logical unit is left inconsistent and unusable. If the copies are not synchronized, you must use the **-force** parameter.

If you run the command while data is in the cache, the system attempts to move the data out of the cache; this process can time out, however.

If any virtual medium errors exist on the volume, the command fails. You can force the deletion by using the **-force** parameter; however, using **-force** can cause data integrity problems.

Note: A virtual medium error occurs when you copy data from one disk (the source) to another (the target). Reading the source indicates that a medium error was found. At that moment, you must have two identical copies of data and you must then simulate a medium error on the target disk. You can simulate a medium error on the target disk by creating a virtual medium error on the target disk.

If FlashCopy mappings or host mappings exist for the volume, the deletion fails unless you use the **-force** parameter. If you use the **-force** parameter, mappings are deleted and the volume is deleted. If any data is not staged in the fast write cache for this volume, the deletion of the volume fails. When the **-force** parameter is specified, any data that is not staged in the fast write cache is deleted. Deleting an image mode volume causes the managed disk that is associated with the volume to be removed from the storage pool. The mode of the managed disk is returned to unmanaged.

If the relationship is in consistent_copying or consistent_stopped state, and the change volume is being used by a Global Mirror relationship that uses multicycling mode, the relationship moves to inconsistent_copying or inconsistent_stopped state.

Note: If the relationship is part of a consistency group, the entire group is affected by this state transition.
The secondary volume becomes corrupted, and inaccessible for host input/output I/O data, if the following conditions are true:

- A changed volume is part of an `idling` relationship.
- The changed volume is being used for secondary protection.
- The background copy process is still migrating the change volume data to the secondary volume.

You must run the **recovervdisk** command to regain access to the volume contents. If all of the following conditions are true, the secondary volume also becomes corrupted:

- The change volume was part of an `idling` relationship.
- The change volume was being used for a Global Mirror relationship that uses multicycling mode.
- The relationship was deleted, but the background copy process continued and is still migrating data to the secondary volume.

In any of these cases, this **recovervdisk** command fails unless you specify the **-force** parameter.

Note:

- The **-force** parameter must be used if **rmvdisk** is specified and rejected if the volume is a change volume for a relationship.
- If the volume is a change volume for a relationship, specifying **rmvdisk** with **-force** removes the change volume from the relationship.

An invocation example

```
rmvdisk -force vdisk5
```

The resulting output:

```
No feedback
```

An invocation example

```
rmvdisk -removehostmappings vdisk3
```

The resulting output:

```
No feedback
```

rmmetadatavdisk

Use the **rmmetadatavdisk** command to detach a file system or remove a block device (that is based on the volume with owner type `host_integration_metadata`) in a configuration node.

Syntax

```
►►── rmmetadatavdisk ── ┬───────────────────┬──────────────────────────────────►◄
                        └─ -ignorevvolsexist ┘
```

Parameters

-ignorevvolsexist
(Optional) Specifies that you want the system to delete the metadata volume (including volumes with owner type `vvol`).

Description

This command removes the metadata volume from a storage pool.

When **-ignorevvolsexist** is specified, only the metadata volume is deleted. Additionally, volumes with owner type `vvol` are not affected.

An invocation example

```
rmmetadatavdisk -ignorevvolsexist
```

The resulting output:

```
No feedback
```

rmvdiskcopy

Use the **rmvdiskcopy** command to remove a volume copy from a volume. This command cannot be used for high availability volumes.

Syntax

```
►►──rmvdiskcopy── ──-copy── ──copy_id── ┬────────┬─ ─┬─ vdisk_name ─┬──────────────►◄
                                        └─-force─┘   └─ vdisk_id ───┘
```

Parameters

-copy *copy_id*
(Required) Specifies the ID of the copy to delete.

-force
(Optional) Forces the deletion of the last synchronized copy of a volume, which deletes the entire volume.

Important: Using the force parameter might result in a loss of access. Use it only under the direction of your product support information.
The parameter also forces the deletion of a nonmirrored volume, a copy that is migrating to image mode, or an image-mode copy that has virtual medium errors.

Important: To prevent an active volume from being deleted unintentionally, you can use a global system setting to enable volume protection (see the **chsystem** command). You can specify a time period for which the volume must be idle before you can delete it. If volume protection is enabled and the time period has not expired, the volume deletion fails even if the **-force** parameter is used."

vdisk_name | *vdisk_id*
(Required) Specifies the volume to delete the copy from. You must specify this parameter last on the command line.

Description

The **rmvdiskcopy** command deletes the specified copy from the specified volume. The command fails if all other copies of the volume are not synchronized; in this case, you must specify the **-force** parameter, delete the volume or more, or wait until the copies are synchronized.

Remember: This command is unsuccessful if:

- Volume protection is enabled
- The last volume copy being deleted has received I/O within the defined volume protection time period

These changes apply to this command only when deleting the last synchronized copy of a volume or removing the entire volume.

An invocation example

This example forces a deletion.

Important: Using the force parameter might result in a loss of access. Use it only under the direction of your product support information.

```
rmvdiskcopy -copy 0 -force 134
```

The resulting output:

```
No feedback
```

An invocation example

This example deletes a mirrored copy from a volume, where 1 is the ID of the copy to delete and vdisk8 is the volume to delete the copy from.

```
rmvdiskcopy -copy 1 vdisk8
```

The resulting output:

```
No feedback
```

rmvdiskaccess

Use the **rmvdiskaccess** command to delete one or more I/O groups from the set of I/O groups in which a volume can be made accessible to hosts.

Syntax

```
►►── rmvdiskaccess ── ──  -iogrp ──┬─ iogrp_id_list ───┬── ──┬─ vdisk_id ───┬──────────────────►◄
                                   └─ iogrp_name_list ─┘     └─ vdisk_name ─┘
```

Parameters

-iogrp *iogrp_id_list* | *iogrp_name_list*
(Required) Specifies a list of I/O groups to remove from the I/O group access set of the volume.

vdisk_id | *vdisk_name*
(Required) Specifies the volume from which to remove access I/O groups.

Description

The **rmvdiskaccess** command removes I/O groups from the volume access set. However, it cannot remove all I/O groups from the access set; a volume must have at least one I/O group in an access set. When an I/O group is removed from the access set, all host mappings created through that I/O group (for the volume) are deleted. Consequently, you cannot access the volume through any related I/O group nodes.

Remember: If an I/O group in the list is not in the access set, no error is generated, but no action is taken for that I/O group.

An invocation example to remove I/O groups 2 and 3 from the volume access set for volume ID 3

```
rmvdiskaccess -iogrp 2:3 3
```

The resulting output:

```
No feedback
```

rmvdiskhostmap

Use the **rmvdiskhostmap** command to delete an existing host mapping the volume is no longer accessible for input/output (I/O) transactions on the given host.

Syntax

```
►►── rmvdiskhostmap ── ── -host ──┬─ host_id ───┬── ──┬─ vdisk_id ───┬──────────────────────────────────►◄
                                  └─ host_name ─┘     └─ vdisk_name ─┘
```

Parameters

-host *host_id* | *host_name*
(Required) Specifies the host that you want to remove from the map with the volume, either by ID or by name.

vdisk_id | *vdisk_name*
(Required) Specifies the name of the volume that you want to remove from the host mapping, either by ID or by name.

Description

This command deletes an existing mapping between the specified volume and the host. This effectively stops the volume from being available for I/O transactions on the given host.

This command also deletes a Small Computer System Interface (SCSI) or persistent reservation that a host has on a volume. Once the reservation is removed, a new host is allowed to access the volume in the future because the original host no longer has access.

Note: The **rmvdiskhostmap** command deletes the host mapping for all I/O groups in the access I/O group set of the volume.

Use caution when you process this command because to the host, it seems as if the volume has been deleted or is offline.

Remember: This command is unsuccessful if:

- Volume protection is enabled
- The host mapping being deleted is mapped to any volume that has received I/O within the defined volume protection time period

An invocation example

```
rmvdiskhostmap -host host1 vdisk8
```

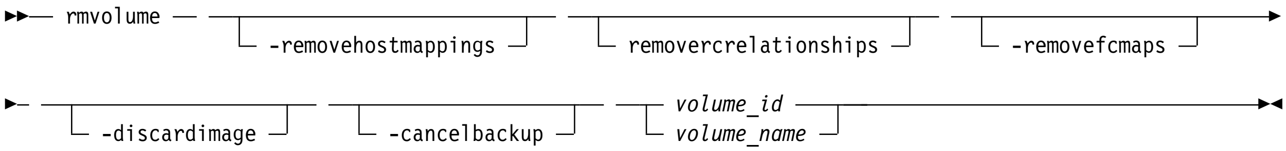
The resulting output:

```
No feedback
```

rmvolume

Use the **rmvolume** command to remove a volume. You can use this command for high availability configurations that include HyperSwap or `stretched` systems.

Syntax



Parameters

-removehostmappings
(Optional) Allows a volume to be deleted even if host mappings are removed if this volume is deleted.

-removercrelationships
(Optional) Allows a volume to be deleted even if it is part of a remote copy relationship.

-removefcmaps
(Optional) Allows a volume to be deleted even if it is part of a FlashCopy mapping and regardless of the state of the mappings. FlashCopy mappings that are `rc_controlled` (for change volumes) require this parameter to be specified to force the deletion of a change volume while it is configured in a remote copy relationship. However, it is recommended to remove the change volume from the relationship before you delete it to avoid data loss. HyperSwap volumes with only the `rc_controlled` FlashCopy mappings for change volumes do not require this parameter to be specified.

-discardimage
(Optional) Allows a volume to be deleted even if data on an image mode copy cannot be made consistent.

Important: Using this parameter might result in data loss. Use it only under the direction of your product support information, or if you are willing to accept the risk of data loss on the volume.

-cancelbackup
(Optional) Allows a volume to be deleted even if a backup operation is in progress.

Important: Using this parameter might result in data loss. Use it only under the direction of your product support information, or if you are willing to accept the risk of data loss on the volume.

volume_id | *volume_name*
(Required) Specifies the volume to be removed.

Description

Use the **rmvolume** command to remove a volume.

For a HyperSwap volume, the `active-active` relationship and the change volumes are also deleted.

An invocation example to remove a volume

```
rmvolume 0
```

The detailed resulting output:

```
No feedback
```

An invocation example to remove a volume with FlashCopy mappings

```
rmvolume -removefcmaps 1
```

The detailed resulting output:

```
No feedback
```

An invocation example to remove a master or auxiliary volume in a Global Mirror relationship with change volumes

```
rmvolume -removercrelationships 6
```

The detailed resulting output:

```
No feedback
```

An invocation example to remove a HyperSwap volume

```
rmvolume myhyperswapvol
```

The detailed resulting output:

```
No feedback
```

An invocation example

```
rmvolume -cancelbackup 1
```

The detailed resulting output:

```
No feedback
```

An invocation example

```
rmvolume -discardimage 1
```

The detailed resulting output:

```
No feedback
```

rmvolumecopy

Use the **rmvolumecopy** command to remove a volume copy from a volume. You can use this command for high availability configurations that include HyperSwap or `stretched` systems.

Syntax

This syntax diagram specifies a volume copy by site.

```
►►── rmvolumecopy ── ── -site ──┬─ site_id ───┬─ ──┬────────────────┬── ──┬───────────────┬── ───────►
                                └─ site_name ─┘    └─ -removefcmaps ─┘    └─ -discardimage ─┘

►──┬─ volume_id ───┬──────────────────────────────────────────────────────────────────────────►◄
   └─ volume_name ─┘
```

This syntax diagram specifies a volume copy by copy ID or storage pool. You must specify either **-copy** or **-pool**.

```
►►── rmvolumecopy ── ──┬────────────────────┬── ──┬───────────────────────────────────────┬── ─────────►
                       └─ -copy ── copy_id ─┘    └─ -pool ──┬─ storage_pool_id ───┬─┘
                                                             └─ storage_pool_name ─┘

►──┬────────────────┬── ──┬───────────────┬── ──┬─ volume_id ───┬──────────────────────────────►◄
   └─ -removefcmaps ─┘    └─ -discardimage ─┘    └─ volume_name ─┘
```

Parameters

-site *site_id* | *site_name*
(Required) Specifies the site that the volume copy is removed from. You cannot specify this parameter if you specify **-copy** or **-pool**.

-pool *storage_pool_id* | *storage_pool_name*
(Optional) Specifies the storage pool that the volume copy is removed from.

-copy *copy_id*
(Optional) Specifies the copy ID for the volume copy to be deleted. The value is 0 or 1. This keyword cannot be specified when **-site** is specified.

Important: If the volume has copies in multiple sites you must specify **-pool** to identify the volume copy to remove.
You must specify either **-copy** or **-pool**.

-removefcmaps
(Optional) Allows a volume copy to be deleted even if it is part of a FlashCopy mapping.

Important: Use this parameter for HyperSwap volumes.
You must specify either **-copy** or **-pool**.

-discardimage
(Optional) Allows a volume copy to be deleted even if data on an image mode copy cannot be made consistent.

Important: Using this parameter might result in data loss. Use it only under the direction of your product support information, or if you are willing to accept the risk of data loss on the volume.

volume_id | *volume_name*
(Required) Specifies the volume ID or name for the volume copy to remove. The value for volume ID must be a number and the value for volume name must be an alphanumeric string.

Description

Use the **rmvolumecopy** command to remove a copy of a volume.

For a HyperSwap volume, the `active-active` relationship and the change volumes are also deleted.

Remember: HyperSwap volumes that are part of a consistency group must be removed from that consistency group before you can remove the last volume copy from that site.

This command fails if a backup operation is in progress.

An invocation example to remove a volume copy at site 1 on a HyperSwap system

```
rmvolumecopy -site 1 0
```

The detailed resulting output:

```
No feedback
```

An invocation example to remove a volume copy when there are two copies in the same storage pool.

```
rmvolumecopy -pool 5 -copy 1 volume5
```

The detailed resulting output:

```
No feedback
```

An invocation example to remove a volume copy with FlashCopy mappings.

```
rmvolumecopy -site 1 -removefcmaps 1
```

The detailed resulting output:

```
No feedback
```

rmvolumegroup

Use the **rmvolumegroup** command to remove a volume group from a system.

Syntax

```
►►── rmvolumegroup ── ┬─ volumegroup_name ─┬─ ──────────────────────────────────────────►◄
                      └─ volumegroup_id ───┘
```

Parameters

volumegroup_name | *volumegroup_id*
(Required) Specifies the volume group name or ID for the volume to remove from the system. The value for the volume group ID must be a number and the value for the volume group name must be an alphanumeric string.

Description

This command removes a volume group from a system.

Note: You cannot delete a volume group if it contains active volumes.

An invocation example

```
rmvolumegroup Vardy1
```

The resulting output:

```
No feedback
```

rmvolumebackupgeneration

Use the **rmvolumebackupgeneration** command to delete a volume backup or cancel a volume snapshot operation that is in progress.

Syntax

```
►►── rmvolumebackupgeneration ──┬──────────────────────────────────┬── ┬─ -generation ── gen_id ─┬──────►◄
                                ├─ -volume ──┬─ volume_name ─┬─────┤   └─ -all ──────────────────┘
                                │            └─ volume_id ───┘     │
                                └─ -uid ── volume_UID ─────────────┘
```

Parameters

-volume *volume_name* | *volume_id*
(Optional) Specifies the volume snapshot by name or ID.

Note: The volume must exist on the local system.
The value for the volume name must be an alphanumeric string and the value for the volume ID must be a number. This parameter is mutually exclusive with **-uid**.

-uid *volume_UID*
(Optional) Specifies the volume snapshot UID. The value for a volume UID must be a number 0 - 32. This parameter is mutually exclusive with **-volume**.

-generation *gen_id*
(Optional) Specifies the snapshot generation to be deleted for the volume. Only a single snapshot generation is deleted. If the specified generation is for a snapshot operation that is in progress, that snapshot operation is canceled. The value for the generation ID must be a number. This command is mutually exclusive with **-all**.

-all
(Optional) Specifies deletion for all volume snapshots (which cancels all snapshot generations as well). This command is mutually exclusive with **-generation**.

Description

This command deletes a volume snapshot or cancels a volume snapshot operation that is in progress.

Note: If the:

- Command completes immediately, the delete operation is performed asynchronously
- Volume has more than one snapshot generation, you cannot delete the most recent snapshot generation

An invocation example

To delete snapshot generation 22 for volume with the UID 600507680CA880DF1800000000000007:

```
rmvolumebackupgeneration -uid 600507680CA880DF1800000000000007 -generation 22
```

The resulting output:

```
No feedback
```

An invocation example

To cancel the current snapshot generation 5 that is in progress for volume vdisk7:

```
rmvolumebackupgeneration -volume vdisk7 -generation 5
```

The resulting output:

```
No feedback
```

An invocation example

To remove all volume snapshots with the name `vdisk10`:

```
rmvolumebackupgeneration -volume vdisk10 -all
```

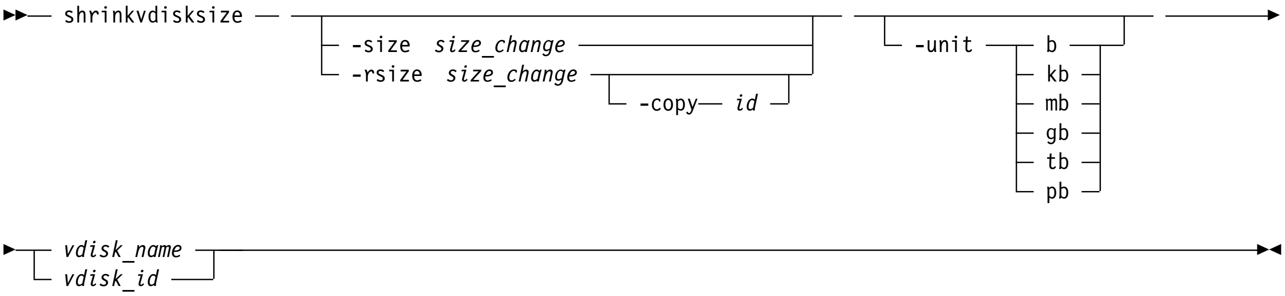
The resulting output:

```
No feedback
```

shrinkvdisksize

Use the **shrinkvdisksize** command to reduce the size of a volume by the specified capacity.

Syntax



Parameters

-size *size_change*
(Optional) Specifies the size reduction (change in size) for the designated volume. The **-size** parameter cannot be used with the **-rsize** parameter. You must specify either **-size** or **-rsize**.

Important: This parameter does reduce the size of a volume (the specified virtual size capacity).

Remember: You cannot use **-size** to resize a thin-provisioned or compressed volume copy that is in a data reduction pool.

-rsize *size_change*
(Optional) Reduces the real size of a thin-provisioned volume by the specified amount. It indicates the change in size as a result of the reduction. Specify the *size_change* value by using an integer. Specify the units for a *size_change* integer by using the **-unit** parameter; the default is MB. You must specify either **-rsize** or **-size**.

Remember: You cannot use **-rsize** to resize a thin-provisioned or compressed volume copy that is in a data reduction pool.

-copy *id*
(Optional) Specifies the copy to change the real capacity for. You must also specify the **-rsize** parameter. If the **-copy** parameter is not specified, all copies of the volume are reduced. This parameter is required if the volume is mirrored and only one copy is thin-provisioned.

-unit b | kb | mb | gb | tb | pb
(Optional) Specifies the data units to be used along with the value that is specified by the **-size** parameter.

vdisk_name | *vdisk_id*
(Required) Specifies the volume that you want to modify, either by ID or by name.

Description

The **shrinkvdisksize** command reduces the capacity that is allocated to the particular volume by the amount that you specify. You cannot shrink the real size of a thin-provisioned volume below its used size. All capacities, including changes, must be in multiples of 512 bytes. An entire extent is reserved even if it is only partially used. The default capacity units are MB. You cannot use **shrinkvdisksize** if the volume is fast formatting.

The command can be used to shrink the physical capacity that is allocated to a particular volume by the specified amount. The command can also be used to shrink the virtual capacity of a thin-provisioned volume without altering the physical capacity that is assigned to the volume. To change the capacity of a non-thin-provisioned disk, use the **-size** parameter. To change the real capacity of a thin-provisioned disk, use the **-rsize** parameter. To change the virtual capacity of a thin-provisioned disk, use the **-size** parameter.

Volumes can be reduced in size, if required.

When the virtual size of a thin-provisioned volume is changed, the warning threshold is automatically scaled to match. The new threshold is stored as a percentage.

To run the **shrinkvdisksize** command on a mirrored volume, all copies of the volume must be synchronized.

Attention: If the volume contains data that is being used, do not shrink the volume without backing up the data first.

The clustered system (system) arbitrarily reduces the capacity of the volume by removing a partial - one or more extents from the ones that are allocated to the volume. You cannot control which extents are removed and so you cannot assume that it is unused space that is removed.

Remember: Before you shrink a volume, validate that the volume is not mapped to any host objects.

You can determine the exact capacity of the source or master volume by issuing the **lsvdisk -bytes** *vdiskname* command. Shrink the volume by the required amount by issuing the **shrinkvdisksize -size** *size_change***-unit** b | kb | mb | gb | tb | pb *vdisk_name* | *vdisk_id* command.

Remember:
1. You cannot resize (shrink) an image mode volume.
2. You cannot resize (shrink) the disk if the volume contains data.
3. You cannot resize (shrink) a volume that is part of a file system.
4. You cannot resize (shrink) volume if that volume is being formatted.
5. You cannot resize (shrink) a volume that is being migrated.
6. You cannot resize (shrink) a volume if cloud snapshot is enabled on that volume.

This command is supported for volumes in Metro Mirror and Global Mirror relationships that are in `consistent_synchronized` state if those volumes are using thin-provisioned or compressed copies.

This command is not supported for volumes:
- In HyperSwap relationships or in Global Mirror relationships that are operating in `cycling` mode.
- In relationships that have a change volume configured.
- That have fully allocated copies.

You must shrink both volumes in a relationship to maintain full operation of the system. To perform this task:

1. Shrink the secondary volume by the required capacity.
2. Shrink the primary volume by the required capacity.

You cannot shrink the virtual size (specified by using the **`-size`** parameter) for thin or compressed volumes in data reduction storage pools. The exception to this is to permit a shrink after you expand the volume if the expanded space has not yet been written to.

An invocation example to decrease the capacity of `vdisk1` by `2` KB

```
shrinkvdisksize -size 2048 -unit b vdisk1
```

The resulting output:

```
No feedback
```

An invocation example to decrease the capacity of `vdisk2` by `100` MB

```
shrinkvdisksize -size 100 -unit mb vdisk2
```

The resulting output:

```
No feedback
```

An invocation example to decrease the real capacity of thin-provisioned `vdisk3` by `100` MB without changing its virtual capacity

```
shrinkvdisksize -rsize 100 -unit mb vdisk3
```

The resulting output:

```
No feedback
```

An invocation example to decrease the real capacity of thin-provisioned VDisk copy ID `1` of mirrored `vdisk3` by `100` MB

```
shrinkvdisksize -rsize 100 -unit mb -copy 1 vdisk4
```

The resulting output:

```
No feedback
```

An invocation example to decrease the virtual capacity of thin-provisioned `vdisk5` by `1` GB without changing its real capacity

```
shrinkvdisksize -size 1 -unit gb vdisk5
```

The resulting output:

```
No feedback
```

splitvdiskcopy

Use the **`splitvdiskcopy`** command to create a separate volume from a synchronized copy of a mirrored volume.

Syntax

```
►►──splitvdiskcopy── ── -copy ── id ── ┬──────────────────────────────┬──────────────────►
                                       └─ -iogrp ──┬─io_group_id───┬─┘
                                                   └─io_group_name─┘
```

```
►─┬──────────────────────────────────────┬─┬─────────────────────────────┬─────────────────►
  └─ -accessiogrp ─┬─ iogrp_id_list ───┬─┘ └─ -node ── ┬─ node_id ───┬─┘
                   └─ iogrp_name_list ─┘               └─ node_name ─┘

►─┬──────────────────────┬─┬──────────────────────────┬─ ─┬─────────────────┬─ ────────────►
  └─ -name ── ── new_name ┘ └─ -cache ─┬─ readwrite ─┬┘   └─ -udid ── ── udid ┘
                                       ├─ readonly ──┤
                                       └─ none ──────┘

►─┬─────────────────┬─ ─┬──────────┬─┬─ vdisk_name ─┬──────────────────────────────────────►◄
  └─ -activeactive ─┘   └─ -force ─┘ └─ vdisk_id ───┘
```

Parameters

-copy *id*
(Required) Specifies the ID of the copy to split.

-iogrp *io_group_id* | *io_group_name*
(Optional) Specifies the I/O group to add the new volume to. The default is the I/O group of the specified volume.

-accessiogrp *iogroup_id_list* | *iogroup_name_list*
(Optional) Specifies which I/O groups provide access to the volume. If the **-accessiogrp** parameter is used, the specified I/O groups provide access even if that set includes either the caching I/O group of the original volume or the caching I/O group of the new volume. If the flag is not specified and the original volume has only its caching I/O group in the set of I/O groups that provide access to the original volume, the new volume is assigned its caching I/O group as the only I/O group that provides access (which might not be the same as caching I/O group of the original volume). Otherwise, the new volume provides access using the same set of I/O groups that are used with the original mirrored volume.

Note: The I/O groups that are specified are not required to include the caching I/O group.

-node *node_id* | *node_name*
(Optional) Specifies the preferred node ID or the name for I/O operations to this volume. You can use the **-node** parameter to specify the preferred access node.

-name *new_name*
(Optional) Assigns a name to the new volume.

-cache readwrite | readonly | none
(Optional) Specifies the caching options for the new volume. (Optional) Specifies the caching options for the volume. Valid entries are:

- readwrite to enable the cache for the volume
- readonly to disable write caching while allowing read caching for a volume
- none to disable the cache mode for the volume

The default is readwrite.

Remember: If you do not specify the **-cache** parameter, the default value (readwrite) is used.

-udid *udid*
(Optional) Specifies the *udid* for the new volume. The *udid* is a required identifier for OpenVMS hosts; no other hosts use this parameter. Supported values are a decimal number 0 - 32767, or a hexadecimal number 0 - 0x7FFF. A hexadecimal number must be preceded by 0x; for example, 0x1234. The default *udid* value is 0.

-activeactive
(Optional) Specifies that an active-active relationship is created between the specified volume and the newly created volume.

-force
(Optional) Allows the split to proceed even when the specified copy is not synchronized, or even when the cache flush is likely to fail. The newly created volume might not be consistent.

Description

The **splitvdiskcopy** command creates a new volume in the specified I/O Group from a copy of the specified volume.

For thin or compressed volumes that are in data reduction storage pools:
- You cannot specify a cache mode of none or readonly
- Specify an I/O group that is different from the current I/O group that is associated with the volume

If the copy that you are splitting is not synchronized, you must use the **-force** parameter. The command fails if you are attempting to remove the only synchronized copy. To avoid command failure, wait for the copy to synchronize or split the unsynchronized copy from the volume by using the **-force** parameter. You can run the command when either volume copy is offline.

For active-active relationships, the existing volume must be in an I/O group with a site name or site ID. The existing volume must also use a storage pool with the same site information. The new volume must be created in an I/O group with a site name or site ID (that is not the same site as the I/O group for the existing volume). The new volume must use a storage pool with the same site name or site ID as the I/O group's site name or site ID. The topology must be hyperswap or the active-active relationship is not allowed.

For active-active relationships, the existing volume must not be the target of a FlashCopy mapping.

You can use this command to partially create a HyperSwap volume and:
1. Configure the access I/O groups for the existing volume to include the new volume's I/O group
2. Create and associate change volumes to the active-active relationship

When these tasks are completed, the active-active relationship can start or resynchronize any regions that are written to on the existing volume. The created relationship uses the existing volume as its master copy, and the new volume as its auxiliary copy.

Note: If the I/O group has enough bitmap space available to allocate for remote copy and the allocated space for the remote copy is not large enough to accommodate the new relationship, space is automatically added. (Remote copy includes Global Mirror, Metro Mirror, and active-active relationships.)

An invocation example for creating a volume with I/O groups 2 and 3 in its I/O group access set

```
splitvdiskcopy -copy 1 -iogrp 2 -node 7 -accessiogrp 2:3 DB_Disk
```

The resulting output:

```
Virtual Disk, copy [1], successfully created.
```

An invocation example

```
splitvdiskcopy -copy 1 vdisk8
```

The resulting output:

```
Virtual Disk, id [1], successfully created.
```

An invocation example

```
splitvdiskcopy -activeactive -iogrp siteB -copy 1 -name siteBvolume siteAvolume
```

The resulting output:

```
Virtual Disk, copy [1], successfully created.
```

Appendix A. Helpful resources and publications

For more general information on how to operate either the IBM FlashSystem 840 or the IBM FlashSystem 900, consult the following related publications and resources:

- IBM Redbooks® publications

Visit the IBM Redbooks website and search for all applicable product information guides for general system information at the following URL:

www.redbooks.ibm.com

- IBM Knowledge Center

Visit the IBM Knowledge Center and use the latest version of documentation for applicable products at the following URL:

https://www.ibm.com/support/knowledgecenter/

Specifically, see the Spectrum Virtualize Software Knowledge Center.

Appendix B. Terminology

Terms that are most commonly used for the command-line interface (CLI) operations.

Table 117 shows the terms and offers a brief description.

Table 117. Abbreviations

Name	Object type
Configuration node or active node	Refers to the node on the system that is currently being managed through the use of the cluster IP. You can run all operations on this node, but run only certain commands on the non-configuration node. See the definition for non-configuration node.
Clustered system (system) or cluster	Referred to as system.
Controller	Refers to a controller that is part of a clustered system.
Host	Refers to a single host.
Host cluster	Refers to a host cluster (which is part of a server that shares a set of volumes).
Host object	Refers to a list of worldwide port names WWPNs and iSCSI names.
IBM FlashCopy mapping	Referred to as `fcmap`.
IBM FlashCopy consistency group	Referred to as `fcconsistgrp`.
IBM HyperSwap mirroring consistency group	Referred to as `hyperswap`.
I/O group	Referred to as `iogrp`.
Managed disk (MDisk)	Referred to as MDisk.
MDisk group or storage pool	Referred to as storage pool.
Metro Mirror or Global Mirror relationship	Referred to as `rcrelationship`.
Metro Mirror or Global Mirror consistency group	Referred to as `rcconsistgrp`.
Node, node canister, enclosure	Refers to node or node canister or enclosure, depending on the system.
Non-configuration node or inactive node	Refers to the node of the I/O group that is not currently being used. Only certain operations can be performed on this node and it is used as a source of redundancy.
Thin-provisioned volume copy	Referred to as `sevdiskcopy`.
VDisk or volume	Referred to as volume.
Volume copy	Referred to as `vdiskcopy`.
Unsupported/unknown object	unknown

Appendix C. HTTP error messages

When there is an issue with the information that you have provided, an error message will appear. Different types of error messages can appear depending on this issue. The only error messages mentioned in this document are HTTP errors. Other error messages are explained in other product documentation, which are listed in Appendix A.

The following HTTP error codes are returned to the user by the RESTful API in response to a problem with the request:

- HTTP Error 400: bad request

  Where the user did not specify a required parameter or gave a parameter value that did not pass the RESTful API's checks.
- HTTP Error 403: forbidden

  Where the user did not send a valid authentication token to interact with the specified URL.
- HTTP Error 404: not found

  Where the user tried to issue a request to a URL that does not exist.
- HTTP Error 405: method not allowed

  Where the user tried to use an HTTP method that is invalid for the specified URL.
- HTTP Error 500: something went wrong on the server

  Where an SVC command error is forwarded from the RESTful API.

Appendix D. Usage examples in Perl

The following Perl example shows how to perform initial set up to start interacting with the system. Following the sample are Perl function calls that import the example file, specify a target IP address, retrieve an authentication token, and finally use the authentication token to carry out an action on a system target.

See Appendix E, "Usage examples in CURL," on page 781 to view more usage examples, which use the **curl** command-line utility.

Note: Values in italics represent values that you supply.

Assume the following file is named `rest_perl.pl`. Included in the file is a function to construct an HTTP request and return the JSON data from the response. These functions can either be imported or copied and pasted at the top of the Perl script.

```
#!/usr/bin/perl -w use strict;
use warnings;
use HTTP::Request; use HTTP::Headers; use LWP::UserAgent; use JSON::PP;
my $ua = LWP::UserAgent->new(ssl_opts => { SSL_verify_mode => 'SSL_VERIFY_NONE' }j;
sub command{
my($host, $target, $method, $header_ref, $data_ref) = @_;
my $data_json = %{$data_ref} ? encode_json($data_ref) : "";
my $request = HTTP::Request->new($method =>
   "https://$host:7443/$target", HTTP::Headers->new(%{$header_ref}), $data_json);
$request->header(Content_Type => 'application/json');
my $response = $ua->request($request);
return decode_json($response->content);
}
```

If the file is being imported, use the following line in another Perl script, or in the Perl debugger:

```
do "rest_perl.pl";
```

Specify the node IP address for the desired target node:

```
$host = "system_IP_address";
```

Provide authentication information to create a token for use in future commands:

```
$token = command($host,
'API_version/auth',
'POST',
{"X-Auth-Password" => "password", "X-Auth-Username" => "username"}, {})->{"token"};
```

To make an array, include the POSTmethod, target, JSON parameters, and the newly created token, as in the following example:

```
$out = command($host,
'1.0/array',
'POST',
{"X-Auth-Token" => $token}, {raid_level => "raid5"});
```

Appendix E. Usage examples in CURL

Several usage examples generated using curl are provided below for reference. For usage examples made using the Perl programming language, see “Getting started” on page 4. Usage examples provided below do not include all actions that might be taken on system objects.

Each curl example takes the following form:

```
curl –k –X POST –H header_1 –H header_2... -d
'JSON' https://system_ip:7443/rest/target
```

Where the following definitions apply:

- POST is the only HTTPS method that the Spectrum Virtualize RESTful API supports.
- Headers are individually-specified HTTP headers (e.g. Content-Type and X-AuthUsername).
- -d is followed by JSON input (e.g. ‘{“raid_level”: “raid5”}’).
- *system_ip* is the IP address that you are sending requests to.
- *target* is the target object of commands, which includes any object IDs, names, and parameters.

Authentication

The following is an example of the correct procedure for authenticating. You authenticate by first producing an authentication token and then using that token in all future commands until the session ends. For more information on authentication, see section 2.3 and section 3.2.

```
curl –k –X POST –H 'Content-Type: application/json' –H 'X-Auth-Username:
superuser' –H 'X-Auth-Password: passw0rd' https://192.168.10.109:7443/rest/auth
```

This yields an authentication token, which can be used for all other commands.

```
{"token": "38823f60c758dca26f3eaa17dc4664964905a6f058ae2ec92e0f0b63fcf08a3e"}
```

Making an array

Most actions must be taken only after authentication, including making an array on the system. The following example demonstrates the use of the token in place of the authentication headers used in the authentication process.

```
curl –k –X POST –H 'Content-Type: application/json' –H 'X-Auth-Token:
38823f60c758dca26f3eaa17dc4664964905a6f058ae2ec92e0f0b63fcf08a3e'
–d '{"level": "raid5", "drive": "6:7:8:9:10", "raid6grp"}' https://192.168.10.109:7443/rest/mkarray
```

Changing array settings

The next example shows how to modify an array member's attributes to change member 24 for new drive 15 by using a distributed rebuild to a rebuild area:

```
curl –k –X POST –H 'Content-Type: application/json' –H 'X-Auth-Token:
38823f60c758dca26f3eaa17dc4664964905a6f058ae2ec92e0f0b63fcf08a3e'
–d '{"member": "24", "newdrive": "15", "immediate": "0"}' https://192.168.10.109:7443/rest/charraymember
```

Removing the array

The final example shows how to remove an array MDisk from the storage pool. For more information on managing the array, see “rmarray” on page 124.

```
curl –k –X POST –H 'Content-Type: application/json' –H 'X-Auth-Token:
38823f60c758dca26f3eaa17dc4664964905a6f058ae2ec92e0f0b63fcf08a3e'
–d '{"mdisk": "6", "mdiskgrp_10"}' https://192.168.10.109:7443/rest/rmarray
```

Appendix F. Accessibility features for the system

Accessibility features help users who have a disability, such as restricted mobility or limited vision, to use information technology products successfully.

Accessibility features

These are the major accessibility features for the system:

- You can use screen-reader software and a digital speech synthesizer to hear what is displayed on the screen. HTML documents are tested by using JAWS version 15.0.
- This product uses standard Windows navigation keys.
- Interfaces are commonly used by screen readers.
- Keys are discernible by touch, but do not activate just by touching them.
- Industry-standard devices, ports, and connectors.
- You can attach alternative input and output devices.

The system online documentation and its related publications are accessibility-enabled. The accessibility features of the online documentation are described in Viewing information in the information center

Keyboard navigation

You can use keys or key combinations for operations and to initiate menu actions that can also be done through mouse actions. You can go to the system online documentation from the keyboard by using the keyboard shortcuts for your browser or screen-reader software. See your browser or screen-reader software Help for a list of keyboard shortcuts that it supports.

IBM and accessibility

See the IBM Human Ability and Accessibility Center for more information about the commitment that IBM has to accessibility.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux and the Linux logo is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other product and service names might be trademarks of IBM or other companies.

Index

Printed in USA