

IBM System Storage SAN Volume Controller



Command-Line Interface User's Guide - Errata

Version 5.1.0
November 4, 2009

SC26-7903-06-Errata

Contents

Introduction	5
Who should use this guide	5
Last Update	5
Change History	5
Chapter 8. Cluster commands	7
addnode	7
chcluster	9
rmpartip	14
Chapter 10. Cluster diagnostic and service-aid commands	17
applysoftware	17
Chapter 12. Virtual disk commands	19
addvdiskcopy	19
chvdisk	23
mkvdisk	28
repairdiskcopy	35
Chapter 14. Managed disk commands	39
applydisksoftware	39
Chapter 16. Metro Mirror and Global Mirror commands	41
rmrcrelationship	41
Chapter 25. Service mode commands	43
cleardumps	43
Chapter 26. Service mode information commands	45
lsnodevpd	45

Introduction

This guide provides errata information that pertains to release 5.1.0 of the *IBM System Storage SAN Volume Controller CLI User's Guide*.

This guide contains the corrections and additions on a per chapter basis. The chapter numbers in this guide correspond directly with the chapter numbers in the *CLI User's Guide* supplied with your SAN Volume Controller.

Who should use this guide

This errata should be used by anyone using the *IBM System Storage SAN Volume Controller Command-Line Interface User's Guide*. You should review the errata contained within this guide and note the details with respect to the copy of the *Command Line Interface User's Guide* supplied with your SAN Volume Controller.

Last Update

This document was last updated: November 4, 2009

Change History

The following revisions have been made to this document:

Revision Date	Sections Modified
November 4, 2009	New publication

Table 1: Change History

Chapter 8. Cluster commands

The following corrections should be noted.

addnode

The following section has been corrected.

Page 48.

You can use the addnode command to add a new (candidate) node to an existing cluster. You can enter this command any time after a cluster has been created. If you are adding a new node to a cluster, you must ensure that the model type of the new node is supported by the SAN Volume Controller software version of the cluster. If the model type is not supported by the cluster software, you must upgrade the cluster to a software version that supports the model type of the new node.

Syntax

```
svctask addnode [-panelname panel_name] [-wwnodename wwnn_arg] [-name new_name_arg] [-iogrp iogroup_name | iogroup_id]
```

Parameters

-panelname *panel_name*

(Required if you do not specify the **-wwnodename** parameter) Specifies the node that you want to add to a cluster by the name that is displayed on the display panel. You cannot use this parameter with the **-wwnodename** parameter.

-wwnodename *wwnn_arg*

(Required if you do not specify the **-panelname** parameter) Specifies the node that you want to add to the cluster by the worldwide node name (WWNN). You cannot use this parameter with the **-panelname** parameter.

-name *new_name_arg*

(Optional) Specifies a name for the node that you want to add to the cluster. You can use this name in subsequent commands to refer to the node, instead of using the node ID. 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 used depends on whether the node is replacing one that has previously been deleted. When a node is deleted, its name will be retained in the iogroup as the failover name of its partner node. If no nodes remain in an

iogroup then all failover names are forgotten. You can only ever store one failover name for each node. If you add a node into an iogroup where a name has been retained as the failover name and do not specify a name in the addnode command, then the retained, failover, name will be assigned to this node. If you do not specify a name and there is no retained, failover, name the name assigned will have the format nodeX, where X is the node ID.

Important: The iSCSI Qualified Name (IQN) for each node is generated using the cluster 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, then adding the node with a different name will change the IQN of this node in the cluster and might require reconfiguration of all iSCSI-attached hosts.

-iogrp *iogroup_name* | *iogroup_id*

(Required) Specifies the I/O group to which you want to add this node.

Description

This command adds a new node to the cluster. You can obtain a list of candidate nodes (those that are not already assigned to a cluster) by typing `svcinfo lsnodecandidate`.

Before you add a node to the cluster, 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 cluster.

- Is the new node being used to replace a failed node in the cluster?
- Does the node being added to the cluster use physical node hardware that has been used as a node in another cluster, and are both clusters 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 `svcinfo lsnode` or the SAN Volume Controller Console to determine the WWNN of the cluster nodes.
2. Shut down all of the hosts that use the cluster, before you add the node back into the cluster.
3. Add the node back to the cluster 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 cluster, you can do the following:
 - a. On all of the hosts that are connected to the cluster, unconfigure the fibre-channel adapter device driver, the disk device driver, and the multipathing driver before you add the node to the cluster.
 - b. Add the node to the cluster and then reconfigure the fibre-channel adapter device driver, the disk device driver, and multipathing driver.

If you are adding a new node to a cluster, take the following actions:

1. Ensure that the model type of the new node is supported by the SAN Volume Controller software version of the cluster. If the model type is not supported by the cluster software, you must upgrade the cluster to a software version 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 has been 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 cluster.

Other considerations when you add a node to a cluster:

When you add a node to the cluster using the `svctask addnode` command or the cluster GUI, you must confirm whether the node has previously been a member of the cluster. 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 cluster using the `svcinfolnode` command.
- If you cannot determine the WWNN of the nodes in the cluster, call the support team to add the node back into the cluster without corrupting the data.

When a node is added to a cluster, it displays a state of adding. It can take as long as 30 minutes for the node to be added to the cluster, particularly if the software version of the node has 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 will be retained in an iogroup as the failover name of its partner node. If no nodes remain in an iogroup then all failover names are forgotten. The `addnode` command will fail if you specify a name that is either an existing node name or a retained, failover, node name. You should specify a different name for the node being added.

An invocation example

```
svctask addnode -wwnodename 5005076801e08b -iogrp io_grp0
```

The resulting output

```
Node, id [6], successfully added
```

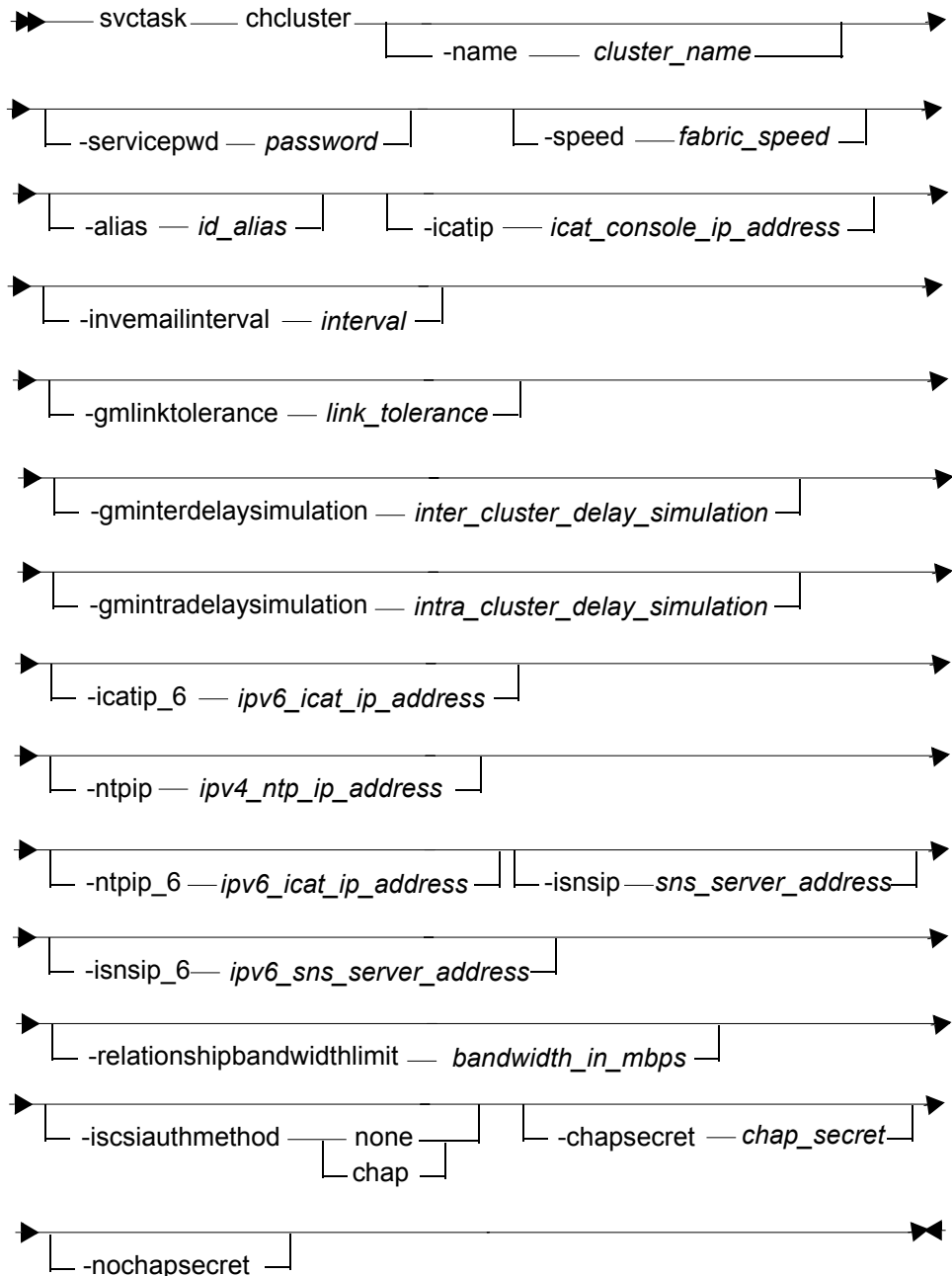
chcluster

The following section has been corrected.

Page 51.

The `chcluster` command modifies the attributes of an existing cluster. You can enter this command any time after a cluster has been created. All the parameters that are associated with this command are optional. However, you must specify one or more parameters with this command.

Syntax



Parameters

-name *cluster_name*

(Optional) Specifies a new name for the cluster.

Important: The iSCSI Qualified Name (IQN) for each node is generated using the cluster and node names. If you are using the iSCSI protocol, changing either name also changes the IQN of all of the nodes in the cluster and might require reconfiguration of all iSCSI-attached hosts.

-servicepwd *password*

(Optional) Specifies a new service user password. You can specify this parameter with or without the password. If the parameter is not followed by a password, you are prompted for the password. When you type the password in response to the prompt, the password is not displayed.

Note: Only a user with administrator authority can change the password.

-speed *fabric_speed*

(Optional) Specifies the speed of the fabric to which this cluster is attached. Valid values are 1 or 2 (GB).

Attention: Changing the speed on a running cluster breaks I/O service to the attached hosts. Before changing the fabric speed, stop I/O from active hosts and force these hosts to flush any cached data by unmounting volumes (for UNIX® host types) or by removing drive letters (for Windows host types). Some hosts might need to be rebooted to detect the new fabric speed.

The fabric speed setting applies only to the SAN Volume Controller model 2145-8F2 in a cluster. The SAN Volume Controller models 2145-8A4, 2145-CF8, 2145-8G4, and 2145-8F4 automatically negotiate the fabric speed on a per-port basis.

-alias *id_alias*

(Optional) Specifies an alternate name that does not change the basic ID for the cluster, but does influence the VDisk_UID of every **vdiskhostmap**, both existing and new. These objects appear to have been created for a cluster whose ID matches the alias. Therefore, changing the cluster alias causes loss of host VDisk access, until each host rescans for VDIs presented by the cluster.

-icatip *icat_console_ip_address*

(Optional) Specifies the new IP address that is used by the cluster. 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.

-invemailinterval *interval*

(Optional) Specifies the interval at which inventory e-mails are sent to the designated e-mail recipients. The interval range is 0 to 15. The interval is measured in days. Setting the value to 0 turns the inventory e-mail notification function off.

-gmlinktolerance *link_tolerance*

(Optional) Specifies the length of time, in seconds, for which an inadequate intercluster link is tolerated for a Global Mirror operation. The parameter accepts values from 60 to 400 seconds in steps of 10 seconds. The default is 300 seconds. You can disable the link tolerance by entering a value of zero (0) for this parameter.

-gminterdelaysimulation *inter_cluster_delay_simulation*

(Optional) Specifies the intercluster delay simulation, which simulates the Global Mirror round trip delay between two clusters, in milliseconds. The default is 0; the valid range is 0 to 100 milliseconds.

-gmintradelaysimulation *intra_cluster_delay_simulation*

(Optional) Specifies the intracluster delay simulation, which simulates the Global Mirror round trip delay in milliseconds. The default is 0; the valid range is 0 to 100 milliseconds.

-icatip_6 *icat_console_ipv6_address*

(Optional) Specifies the new IPv6 address that is used by the cluster. If you specify this parameter, it overwrites any existing **-icatip** address. The format of the IPv6 address must be one of the following:

- 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 cluster to immediately start using that NTP server as its time source. To stop using the NTP server as a time source, invoke the **-ntpip** parameter with a zero address, as follows:

```
svctask chcluster -ntpip 0.0.0.0
```

-ntpip_6 *ipv6_ntp_ip_address*

(Optional) Specifies the IPv6 address for the NTP server. Configuring an NTP server address causes the cluster to immediately start using that NTP server as its time source. To stop using the NTP server as a time source, invoke the **-ntpip_6** parameter with a zero address, as follows:

```
svctask chcluster -ntpip_6 0::0
```

-isnsip *sns_server_address*

(Optional) Specifies the IPv4 address for the iSCSI storage name service (SNS).

To stop using the configured IPv4 iSCSI SNS server, invoke the **-isnsip** parameter with a zero address, as follows:

```
svctask chcluster -isnsip 0.0.0.0
```

-isnsip_6 *ipv6_sns_server_address*

(Optional) Specifies the IPv6 address for the iSCSI SNS.

To stop using the configured IPv6 iSCSI SNS server, invoke the **-isnsip_6** parameter with a zero address, as follows:

```
svctask chcluster -isnsip_6 0::0
```

-relationshipbandwidthlimit *bandwidth_in_mbps*

(Optional) Specifies the new background copy bandwidth in megabytes per second (MBps), from 1 - 1000. The default is 25 MBps. This parameter operates cluster-wide and defines the maximum background copy bandwidth that any relationship can adopt. The existing background copy bandwidth settings defined on a partnership continue to operate, with the lower of the partnership and VDisk rates attempted.

Note: Do not set this value higher than the default without establishing that the higher bandwidth can be sustained.

-iscsiauthmethod **none** | **chap**

(Optional) Sets the authentication method for the iSCSI communications of the cluster. The **iscsiauthmethod** value can be **none** or **chap**.

-chapsecret *chap_secret*

(Optional) Sets the Challenge Handshake Authentication Protocol (CHAP) secret to be used to authenticate the cluster via 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. This parameter is not allowed if the **chapsecret** parameter is specified.

Description

This command modifies specific features of a cluster. Multiple features can be changed by issuing a single command.

Using the **-ntpip** or **-ntpip_6** parameter allows the cluster to use an NTP server as an outside time source. The cluster 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's clock. In the NTP mode, the `svctask setclustertime` command is disabled.

All command parameters are optional; however, you must specify at least one parameter.

Modifying a password: To change the service user password, issue the **svctask chcluster -servicepwd *password*** command.

Note: If you do not want the password to display as you enter it on the command line, omit the new password. The command line tool then prompts you to enter and confirm the password without the password being displayed.

Use the **svctask chclusterip** command to modify the cluster IP address and service IP address.

An invocation example

```
svctask chcluster -ntpip 9.20.165.16
```

The resulting output

No feedback

rmportip

The following section has been corrected.

Page 67.

The **rmportip** command removes an iSCSI IP address from a node ethernet port.

Syntax

```
➡— svctask— rmportip —————➡  
          |———— -failover————|       |———— -ip_6 ————|  
➡— -node — node_name ————— port_id —————➡  
          |———— node_id —————|
```

Parameters

-failover

(Optional) Specifies that the failover IP address information be removed for the specified port.

-ip_6

(Optional) Specifies that the IPv6 address be removed for the specified port. If this parameter is not used, the 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 or 2) to apply changes to.

Description

This command removes an IPv4 or IPv6 address from an ethernet port of a node.

An invocation example for IPv4

```
svctask rmpoortip -node 1 1
```

The resulting output

No feedback

An invocation example for IPv6

```
svctask rmpoortip -node 1 -ip_6 2
```

The resulting output

No feedback

Chapter 10. Cluster diagnostic and service-aid commands

The following correction should be noted.

applysoftware

The following section has been corrected.

Page 79

The applysoftware command upgrades the cluster to a new level of software.

Syntax

```
▶▶ svctask — applysoftware — -file — filename_arg — [ -force ] ▶▶
```

OR

```
▶▶ svctask — applysoftware — -abort — [ -force ] ▶▶
```

Parameters

-file *filename_arg*

(Required for performing an upgrade) Specifies the file name of the new software package to be applied.

-abort

(Required for stopping an upgrade) Specifies that a stalled upgrade should be stopped, returning the cluster to the original software level.

Note: The **force** parameter can be used with the **abort** parameter. If one or more nodes are offline, you must use the **force** parameter with the **abort** parameter.

-force

(Optional) Specifies that the upgrade or abort should proceed even if there are non-redundant nodes in the cluster.

Note: Using this option may result in a loss of access

Description

This command starts the upgrade process of the cluster to a new level of SAN Volume Controller software. The `applysoftware` command applies a level of software to the node in both service and nonservice modes. In service mode, the `applysoftware` command is applied to the specific node. In nonservice mode, the `applysoftware` command is applied to the entire cluster.

The software package as specified by the file name must first be copied onto the current configuration node in the `/home/admin/upgrade` directory. You can use the PuTTY secure copy (`scp`) application to copy the file.

The command completes as soon as the upgrade process has successfully begun.

The command fails and the upgrade package is deleted if:

- The given package fails an integrity check due to corruption.
- Any node in the cluster has a hardware type not supported by the new software.
- The new software level does not support upgrades from the currently installed software.
- The software level of a remote cluster is incompatible with the new software.
- There are any VDIs that are dependent on the status of a node.

Note: The `force` parameter can be used to override this if you are prepared to lose access to data during the upgrade. Before proceeding, use the `svcinfo lsnodedependentvdisks` command to list the node-dependent VDIs at the time the command is run. If the command returns an error, you must move your quorum disks to MDIs that are accessible through all nodes. Rerun the command until no errors are returned.

The actual upgrade completes asynchronously.

The `svcinfo lssoftware` command allows you to view the contents of the `/home/admin/upgrade` directory.

An invocation example

```
svctask applysoftware -file softwareupdate
```

The resulting output

No feedback

Chapter 12. Virtual disk commands

The following corrections should be noted.

addvdiskcopy

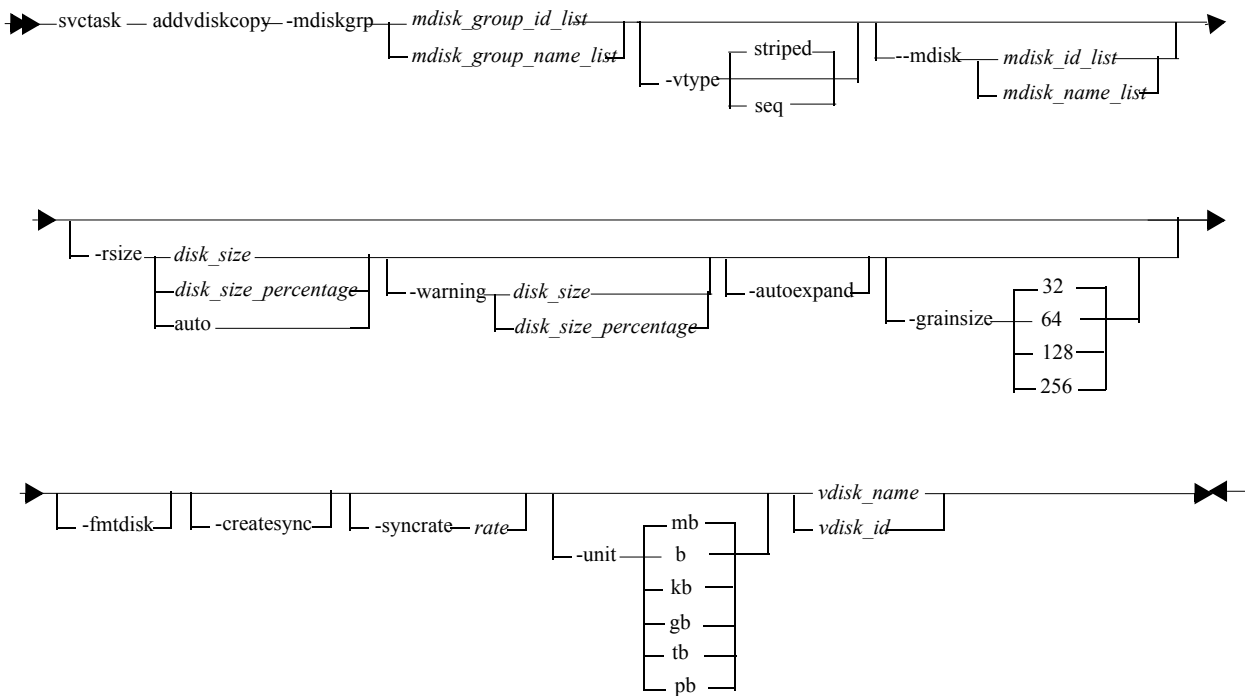
The following section has been corrected.

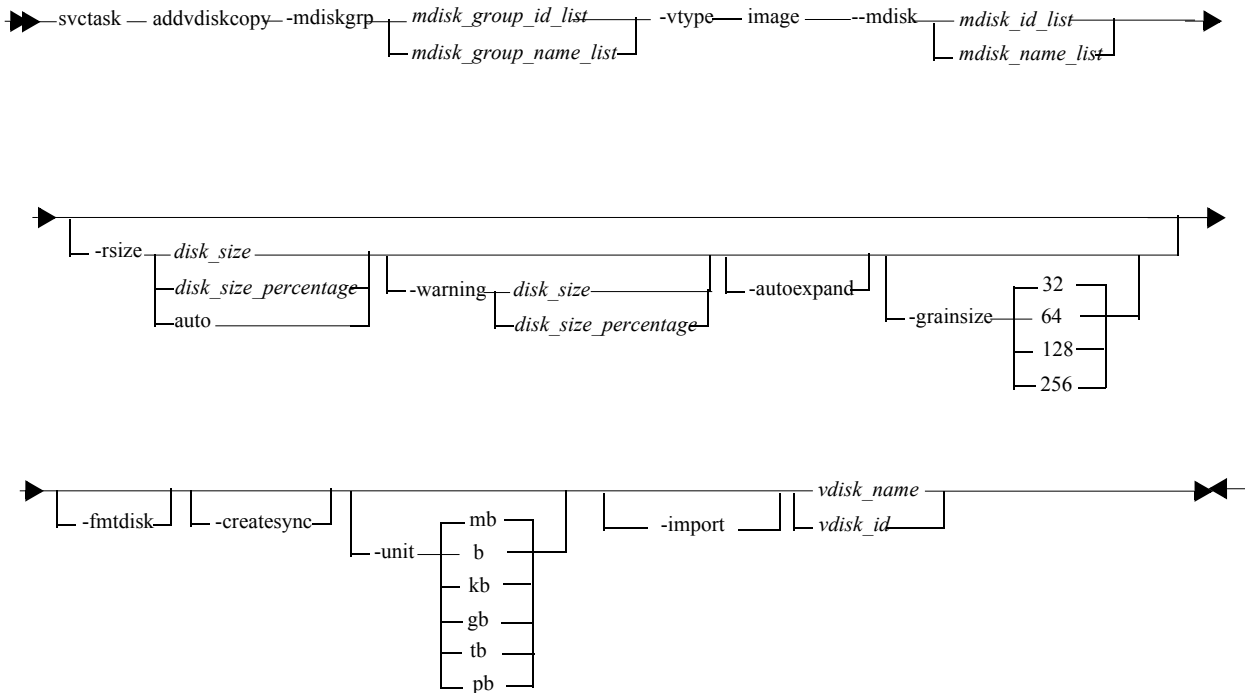
Page 97.

The **addvdiskcopy** command adds a copy to an existing VDisk, which changes a nonmirrored VDisk into a mirrored VDisk.

Note: The first syntax diagram depicts the addition of a sequential or striped mode virtual disk. The second syntax diagram depicts the addition of an image mode virtual disk.

Syntax





Parameters

-mdiskgrp *mdisk_group_id_list* | *mdisk_group_name_list*

(Required) Specifies the managed disk groups to use to create copies for the virtual disk. You must specify a group for each copy that is being added.

-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 VDisk. The default virtualization type is **striped**.

-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 *rate*

(Optional) Specifies the copy synchronization rate. A value of zero (**0**) prevents synchronization. The default value is **50**. For the supported **-syncrate** values and their corresponding rates, see Table 7 on page 23.

Note: If the VDisk copies will be stored on solid state drives, the syncrate must be set to 80 or more. A higher value will be required if there is a very high write rate to the VDisk.

-createsync

(Optional) Suppresses the synchronization of the new VDisk 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. To avoid data loss or read stability loss, use this parameter only for a primary copy that has been formatted and not written to, and with the **-fmtdisk** parameter.

-fmtdisk

(Optional) Formats a sequential or striped mode copy. You must also specify the **-createsync** parameter, which labels the formatted copy as identical to the primary copy. The **-fmtdisk** parameter causes the VDisk to go offline until new VDisk copy formatting completes. To query the formatting progress, use the **lsvdiskprogress** command.

-rsize *disk_size* | *disk_size_percentage%* | **auto**

(Optional) Makes the copy space-efficient and specifies the real size of the copy. Specify the *disk_size* | *disk_size_percentage* value 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 VDisk copy that uses the entire size of the MDisk; if you specify the **-rsize auto** option, you must also specify the **-vtype image** option.

-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 space-efficient 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 virtual disk size. If **-autoexpand** is enabled, the default value for **-warning** is 80% of the virtual disk 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 space-efficient copies automatically expand their real capacities by allocating new extents from their managed disk group. If the **-autoexpand** parameter is specified, the **-rsize** parameter specifies a capacity that is reserved by the copy.

This protects the copy from going offline when its managed disk group 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 space-efficient VDisk. The default is 32 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 space-efficient volume into the cluster. Requires that the **-rsize** and **-vtype image** parameters also be specified.

vdisk_name | *vdisk_id*

(Required) Specifies the virtual disk to add the VDisk copy to, either by ID or by name.

Description

The **addvdiskcopy** command adds a copy to an existing VDisk, which changes a nonmirrored VDisk into a mirrored VDisk. Use the **-mdiskgrp** parameter to specify the managed disk group that will provide storage for the copy; the **svcinfo lsmdiskgrp** command lists the available managed disk groups and the amount of available storage in each group.

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 virtual disk using extents from the given managed disk (assuming there are enough free extents on the managed disk).

striped

This 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 managed disk group are used to create the virtual disk. The striping is at an extent level; one extent from each managed disk in the group is used. For example, a managed disk group 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 can be two or more managed disks from the same managed disk group. 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 virtual disks to be created when a managed disk already has data on it, perhaps from a previrtualized subsystem. When an image mode virtual disk is created, it directly corresponds to the (previously unmanaged) managed disk that it was created from; therefore, virtual disk 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 cluster. After it is under the control of the cluster, you can migrate the virtual disk from the single managed disk. When it is migrated, the virtual disk is no longer an image mode virtual disk.

You can add image mode VDIs to an already populated MDisk group with other types of VDIs, such as a striped or sequential.

Note: An image mode copy must be at least as large as the VDisk that it is being added to, but any capacity beyond the size of the VDisk is not accessible.

The command returns the ID of the newly created VDisk copy.

Table 6 provides the relationship of the *rate* value to the data copied per second.

Table 6. Relationship between the rate value and the data copied per second

User-specified <i>rate</i> 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

```
svctask addvdiskcopy -mdiskgrp 0 vdisk8
```

The resulting output

```
Vdisk [8] copy [1] successfully created
```

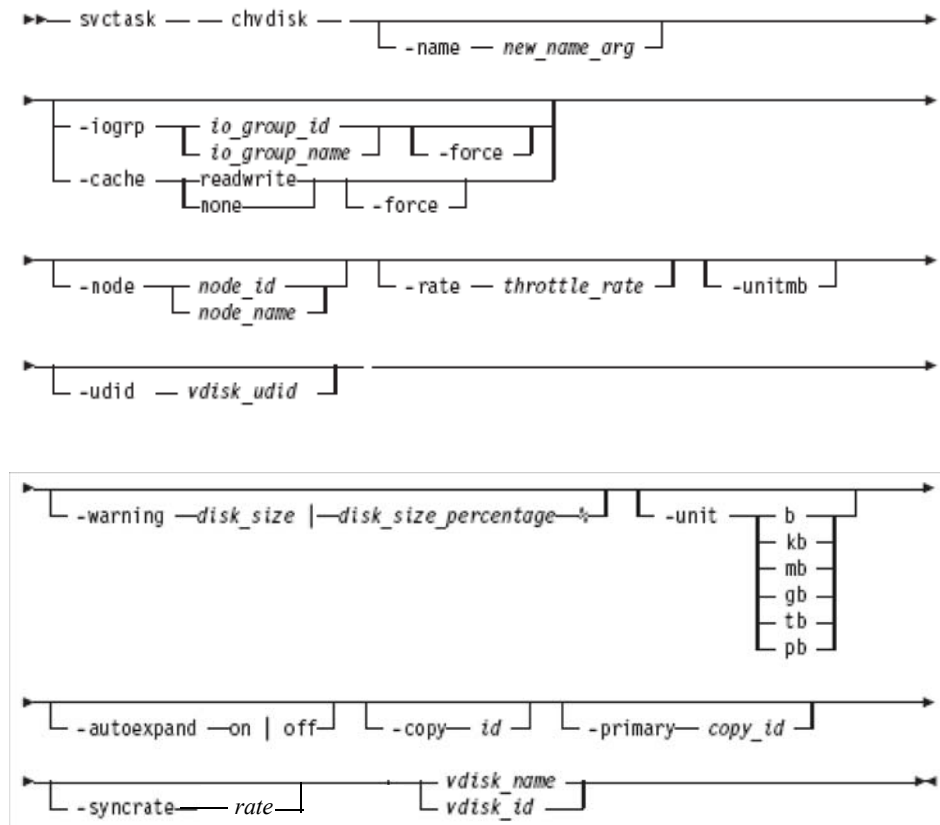
chvdisk

The following section has been corrected.

Page 100.

The `chvdisk` command modifies the properties of a virtual disk, such as the disk name, I/O group, I/O governing rate, or unit number.

Syntax



Parameters

-name *new_name_arg*

(Optional) Specifies a new name to assign to the virtual disk. You cannot use this parameter with the **-iogrp**, **-rate**, **-node**, or **-udid** parameters. This parameter is required if you do not use the **-iogrp**, **-rate**, or **-udid** parameter.

-iogrp *io_group_id | io_group_name*

(Optional) Specifies a new I/O group to move the virtual disk to, by I/O group ID or I/O group name. You can use the **-node** parameter with the **-iogrp** parameter to specify a preferred node for the specified VDisk.

Notes:

- 1.If the VDisk has a mapping to any hosts, it is not possible to move the VDisk to an I/O group, unless all of those hosts are associated with the new I/O group.
- 2.This parameter can fail if there is not enough space to allocate bitmaps for a mirrored VDisk in the target I/O group.
- 3.This parameter can fail if any copy is not synchronized. The **-force** parameter can be used to force the move, but this resynchronizes the VDisk.

4.If the VDisk is offline, use one of the `recovervdisk` commands to recover the VDisk and bring it back online. Beginning with SAN Volume Controller version 4.3.1, use of the recovery I/O group is not required.

-cache readwrite | none

(Optional) Specifies the caching options for the VDisk. Valid entries are **readwrite**, to enable the cache for the VDisk, or **none**, to disable the cache mode for the VDisk.

-force

(Optional) The **force** parameter can only be used for changing the I/O group of a VDisk or the caching mode. Use the **force** parameter with the **iogrp** parameter to force the VDisk to be removed from an I/O group. Use the **force** parameter with the **cache** parameter to specify that you want the system to change the cache mode of the VDisk even if the I/O group is offline. This option overrides the cache flush mechanism.

Attention:

1.If the **force** parameter is used for changing the caching mode or I/O group of a VDisk, the contents of the cache are discarded and the VDisk might be corrupted by the loss of the cached data. This could occur if the cluster is able to destage all write data from the cache or not. The **force** parameter should be used with caution.

2.If the **force** parameter is used to move a VDisk that has out-of-sync copies, a full resynchronization is required.

-rate throttle_rate [-unitmb]

(Optional) Specifies the I/O governing rate for the VDisk, which caps the amount of I/O that is accepted. The default *throttle_rate* units are I/Os. To change the *throttle_rate* units to megabytes per second (MBps), specify the **-unitmb** parameter. The governing rate for a virtual disk can be specified by I/Os or by MBps, but not both. However, you can set the rate to I/Os for some virtual disks and to MBps for others.

You cannot use this parameter with the **-name**, **-iogrp**, **-node**, 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 from 0 to 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**, **-iogrp**, **-node**, or **-rate** parameters.

-warning disk_size | disk_size_percentage%

(Optional) Generates a warning when the used disk capacity on the space-efficient 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 virtual disk 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.

-autoexpand on | off

(Optional) Specifies whether space-efficient VDisk copies automatically expand their real capacities by allocating new extents from their managed disk group. To use this parameter, the VDisk must be space-efficient.

-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 VDisk is mirrored and only one VDisk copy is space-efficient. If both copies are space-efficient 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 only takes effect 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.

-syncrate rate

(Optional) Specifies the copy synchronization rate, as a percentage of the peak synchronization rate. A value of zero (**0**) prevents synchronization.

Note: If the VDisk copies will be stored on solid state drives, the syncrate must be set to 80 or more. A higher value will be required if there is a very high write rate to the VDisk.

-node node_id | node_name

(Optional) Specifies a preferred node for the specified VDisk. When using this parameter, you must also specify the **-iogrp** parameter. You cannot use this parameter with the **-name**, **-rate**, or **-udid** parameters.

vdisk_name | vdisk_id

(Required) Specifies the virtual disk to modify, either by ID or by name.

Description

The `chvdisk` command modifies a single property of a virtual disk (VDisk). To change the VDisk name and modify the I/O group, for example, you must issue the command twice.

You can specify a new name or label. You can use the new name subsequently to refer to the virtual disk. To specify a preferred node for the VDisk, use the **-node node_id | node_name** parameter.

You can change the I/O group with which this virtual disk is associated. However, to change the I/O group, you must first flush the cache within the nodes in the current I/O group to ensure that all data is written to disk. Ensure that you suspend I/O operations at the host level before you perform this operation.

Attention:

1. Do not move a VDisk to an offline I/O group under any circumstance. To avoid any data loss, you must ensure that the I/O group is online before you move the VDIs.
2. Do not move an offline VDisk to the recovery I/O group. Beginning with SAN Volume Controller version 4.3.1, use of the recovery I/O group is not required. Instead, use one of the `recovervdisk` commands to recover the VDisk and bring it back online.

You can set a limit on the amount of I/O transactions that is accepted for this virtual disk. 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 virtual disk 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 only happen when byte units (**-b**) are used. The default capacity is in MB.

When the virtual disk is created, there is no throttling applied to it. Using the **-rate** parameter can change this. To change the virtual disk back to an unthrottled state, specify 0 (zero) with the **-rate** parameter.

You can migrate a VDisk to a new I/O group to manually balance the workload across the nodes in the cluster. You might end up with a pair of nodes that are overworked and another pair that are underworked. Use the following procedure to migrate a single VDisk to a new I/O group. Repeat for other VDIs as required.

Attention: This is a disruptive procedure. Access to the VDisk is lost while you follow this procedure.

Ensure that when you migrate a VDisk to a new I/O group, you quiesce all I/O operations for the VDisk. Determine the hosts that are using this VDisk. Stop and delete any FlashCopy mappings or Metro or Global Mirror relationships that use this VDisk. To check if the VDisk is part of a relationship or mapping, issue the `svcinfolsvdisk vdiskname | id` command, where *vdiskname* | *id* is the name or ID of the VDisk.

Look for the **FC_id** and **RC_id** fields. If these are not blank, the VDisk is part of a mapping or relationship. See the FlashCopy commands or Metro Mirror and Global Mirror commands for details on how to stop or delete the mapping or relationship. Issue the following command to migrate the VDisk:

```
svctask chvdisk -iogrp newiogrpname|id vdiskname|id
```

Follow the procedure to discover the new vpaths and to check that each vpath is presenting the correct number of paths. Refer to Multipath Subsystem Device Driver (SDD) documentation for details on how to dynamically reconfigure SDD for the given host operating system.

Note: The command fails if you attempt to change the primary copy of a mirrored VDisk while the repairvdiskcopy -resync command is running.

An invocation example

```
svctask chvdisk -rate 2040 -unitmb 6
```

The resulting output

No feedback

mkvdisk

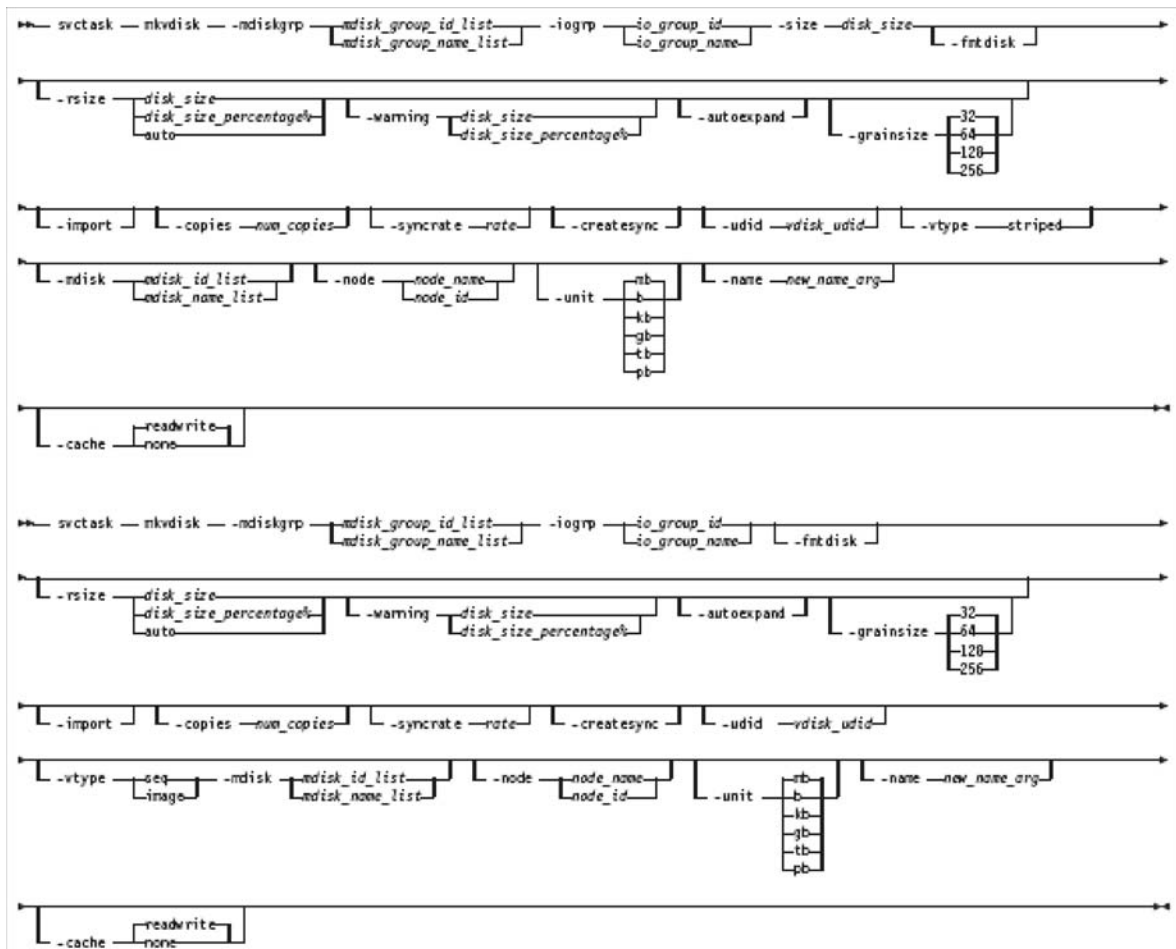
The following section has been corrected.

Page 106.

The **mkvdisk** command creates sequential, striped, or image mode virtual disk objects. When they are mapped to a host object, these objects are seen as disk drives with which the host can perform I/O operations.

Note: The first syntax diagram depicts the creation of a **striped** mode virtual disk. The second syntax diagram depicts the creation of a **sequential** or **image** mode virtual disk.

Syntax



Parameters

-mdiskgrp *mdisk_group_id_list* | *mdisk_group_name_list*

(Required) Specifies one or more managed disk groups to use when you are creating this virtual disk. If you are creating multiple copies, you must specify one managed disk group per copy. The primary copy is allocated from the first managed disk group in the list.

-iogrp *io_group_id* | *io_group_name*

(Required) Specifies the I/O group (node pair) with which to associate this virtual disk.

-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**).

-size *disk_size*

(Required for sequential [**seq**] or **striped** VDisk creation) (Optional for **image** VDisk creation) Specifies the capacity of the virtual disk, 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, which 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.

-rsize *disk_size* | *disk_size_percentage%* | **auto**

(Optional) Makes the VDisk space-efficient; otherwise, the VDisk is fully allocated. Specify the *disk_size* | *disk_size_percentage* value using an integer, or an integer immediately followed by the percent character (%). Specify the units for a *disk_size* integer using the **-unit** parameter; the default is MB. The **-rsize** value can be greater than, equal to, or less than the size of the VDisk. The **auto** option creates a VDisk copy that uses the entire size of the MDisk; if you specify the **-rsize auto** option, you must also specify the **-vtype image** option.

-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 VDisk copies. A warning is generated when the used disk capacity on the space-efficient 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 virtual disk size. If **-autoexpand** is enabled, the default value for **-warning** is 80% of the virtual disk capacity. If **-autoexpand** is not enabled, the default value for warning is 80% of the real capacity. To disable warnings, specify **0**.

-autoexpand

(Optional) Specifies that space-efficient copies automatically expand their real capacities by allocating new extents from their managed disk group. 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. This protects the copy from going offline when its managed disk group runs out of space by having the managed disk group to consume this reserved space first.

The parameter has no immediate effect on **image** mode copies. However, if the image mode copy is subsequently migrated to managed mode, the copy is then automatically expanded.

-grainsize 32 | 64 | 128 | 256

(Optional) Sets the grain size (KB) for a space-efficient VDisk. This parameter also requires that the **-rsize** parameter be specified. The default is 32 KB. If you are using the space-efficient VDisk in a FlashCopy map, use the same grain size as the map grain size for best performance. If you are using the space-efficient VDisk directly with a host system, use a small grain size.

-import

(Optional) Imports a space-efficient VDisk from the Mdisk. This parameter also requires that the **-rsize** parameter be specified.

-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 VDisk. The default value is 1.

-syncrate *rate*

(Optional) Specifies the copy synchronization rate. A value of zero (0) prevents synchronization. The default value is **50**. For the supported **-syncrate** values and their corresponding rates, see Table 7 on page 34.

Note: If the VDisk copies will be stored on solid state drives, the syncrate must be set to 80 or more. A higher value will be required if there is a very high write rate to the VDisk.

-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 VDisk is not required.

-fmtdisk

(Optional) Specifies that the virtual disk be formatted before it can be used. The **-fmtdisk** parameter formats (sets to all zeros) the extents that make up this VDisk after it is created. If this parameter is used, the command completes asynchronously; you can query the status using the **svcinfo lsvdiskprogress** command.

The **-fmtdisk** parameter is not required when creating space-efficient virtual disks. Space-efficient VDIsks return zeros for extents that have not been written to.

The **-fmtdisk** parameter synchronizes mirrored copies by default.

Note: You cannot specify this parameter with the **-vtype image** parameter.

-vtype seq | striped | image

(Optional) Specifies the virtualization type. When creating sequential or image mode VDIsks, you must also specify the **-mdisk** parameter. The default virtualization type is striped.

-node *node_id* | *node_name*

(Optional) Specifies the preferred node ID or the name for I/O operations to this virtual disk. You can use the **-node** parameter to specify the preferred access node.

Note: This parameter is required for the subsystem device driver (SDD). The cluster chooses a default if you do not supply this parameter.

-unit b | kb | mb | gb | tb | pb

(Optional) Specifies the data units to use in conjunction with the capacity that is specified by the **-size** and **-rsize** parameters.

-mdisk *mdisk_id_list* | *mdisk_name_list*

(Optional) Specifies one or more managed disks. For sequential and image mode VDIs, the number of MDIs must match the number of copies. For sequential mode VDIs, each MDI must belong to the specified MDI group. For striped VDIs, you cannot specify the **-mdisk** parameter if the **-copies** value is greater than 1. When creating a single copy striped VDI, you can specify a list of MDIs to stripe across.

-name *new_name_arg*

(Optional) Specifies a name to assign to the new virtual disk.

-cache *readwrite* | *none*

(Optional) Specifies the caching options for the VDI. Valid entries are **readwrite** or **none**. The default is **readwrite**. If you do not specify the **-cache** parameter, the default value (**readwrite**) is used.

Description

This command creates a new virtual disk object. You can use the command to create a variety of types of virtual disk objects, making it one of the most complex commands.

You must decide which managed disk group or groups provide the storage for the VDI. Use the **svcinfolsmdiskgrp** command to list the available managed disk groups and the amount of free storage in each group. If you are creating a VDI with more than one copy, each MDI group that you specify must have enough space for the size of the VDI.

Choose an I/O group for the VDI. This determines which nodes in the cluster process the I/O requests from the host systems. If you have more than one I/O group, ensure that you distribute the VDIs between the I/O groups so that the I/O workload is shared evenly between all nodes. Use the **svcinfolsiogrp** command to show the I/O groups and the number of virtual disks that are assigned to each I/O group.

Note: It is normal for clusters with more than one I/O group to have MDI groups that have VDIs in different I/O groups. FlashCopy processing can make copies of VDIs whether the source and target VDIs are in the same I/O group. If, however, you plan to use intracluster Metro or Global Mirror operations, ensure that both the master and auxiliary VDI are in the same I/O group.

Specify the virtualization type using the **-vtype** parameter; the supported types are sequential (**seq**), **striped**, and **image**.

sequential (seq)

This virtualization type creates the virtual disk using sequential extents from the specified MDI (or MDIs, if creating multiple copies). The command fails if there are not enough sequential extents on the specified MDI.

striped

This is the default virtualization type. If the **-vtype** parameter is not specified, **striped** is the default; all managed disks in the managed disk group are used to create the virtual disk. The striping is at an extent level; one extent from each managed disk in the group is used. For example, a managed disk group 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 can be two or more managed disks from the same managed disk group. 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 virtual disks to be created when a managed disk already has data on it, perhaps from a previrtualized subsystem. When an image mode virtual disk is created, it directly corresponds to the (previously unmanaged) managed disk that it was created from. Therefore, with the exception of space-efficient image mode VDIs, virtual disk 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 cluster. After it is under the control of the cluster, you can migrate the virtual disk from the single managed disk. When it is migrated, the virtual disk is no longer an image mode virtual disk.

You can add image mode VDIs to an already populated MDisk group with other types of VDIs, such as a striped or sequential.

Note: An image mode VDisk must be 512 bytes or greater. At least one extent is allocated to an image mode VDisk.

You must use the **-mdisk** parameter to specify an MDisk that has a mode of unmanaged. The **-fmtdisk** parameter cannot be used to create an image mode VDisk.

Note: If you create a mirrored VDisk from two image mode MDisks without specifying a **-capacity** value, the capacity of the resulting VDisk is the smaller of the two MDisks, and the remaining space on the larger MDisk is not accessible.

The command returns the IDs of the newly created VDisk.

Attention:

1. Do not create a VDisk in an offline I/O group. You must ensure that the I/O group is online before you create a VDisk to avoid any data loss. This

applies in particular to recreating VDisks that are assigned the same object ID.

2. To create an image mode disk, you must already have a quorum disk in the cluster because an image mode disk cannot be used to hold quorum data. See “Creating a quorum disk” in the *IBM System Storage SAN Volume Controller Software Installation and Configuration Guide* for more details.
3. The command fails if either limit of 2048 VDisks per I/O Group or 8192 VDisk copies per cluster is reached.

Table 7 provides the relationship of the *rate* value to the data copied per second.
Table 7. Relationship between the rate value and the data copied per second

User-specified <i>rate</i> 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

```
svctask 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 VDisk

```
svctask 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 VDisk
svctask mkvdisk -mdiskgrp Group0 -size 0 -unit kb
-iogrp 0 -vtype striped -mdisk mdisk1 -node 1 -udid 1234

The resulting output
Virtual Disk id [2], successfully created

An invocation example for creating a space-efficient VDisk
svctask 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 mirrored image-mode VDisk
svctask 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 VDisk
svctask mkvdisk -iogrp 0 -mdiskgrp 0:1 -size 500 -copies 2

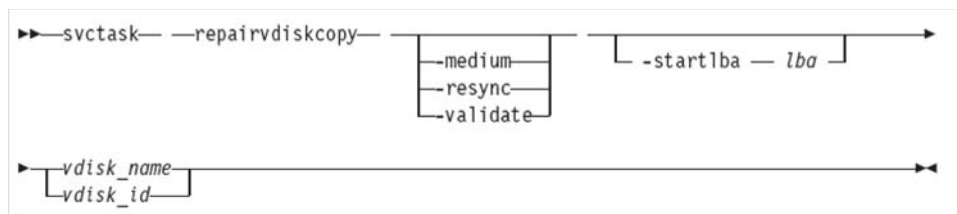
The resulting output
Virtual Disk id [5], successfully created

repairvdiskcopy

*The following section has been corrected.
Page 117.*

The **repairvdiskcopy** command detects and optionally, corrects any VDisk copies that are not identical.

Syntax



Parameters

-medium

(Optional) Converts sectors that contain different contents into virtual medium errors on the specified VDisk. This parameter cannot be used with the **-validate** and **-resync** parameters; you must enter one of the three parameters.

-resync

(Optional) Corrects sectors that contain different contents by copying contents from the primary VDisk copy to other copies on the specified VDisk. This parameter cannot be used with the **-medium** and **-validate** parameters; you must enter one of the three parameters.

-validate

(Optional) Reports the first difference found on synchronized online copies of the specified VDisk, 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.

vdisk_name | vdisk_id

(Required) Specifies the virtual disk to repair. You must specify this parameter last on the command line.

Description

The **repairvdiskcopy** command detects and optionally, corrects any VDisk copies that are not identical. The results are logged to the SAN Volume Controller error log. The **-validate** parameter compares synchronized online copies of the specified VDisk. 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 VDisk copies. You must specify only one of the three parameters.

Attention:

1. Before you run the **repairvdiskcopy** command, ensure that all VDisk copies are synchronized.
2. Only one **repairvdiskcopy** command can run on a VDisk 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 VDisk 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 - 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 VDisk copies are not identical.

During **repairvdiskcopy** command operation, the VDisk 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 VDisk that is being repaired. To suspend the repair process, set the synchronization rate of the VDisk to **0** using the **chvdisk** command.

An invocation example

```
svctask repairvdiskcopy -resync vdisk8
```

The resulting output

```
No feedback
```

Chapter 14. Managed disk commands

The following correction should be noted.

applymdisksoftware

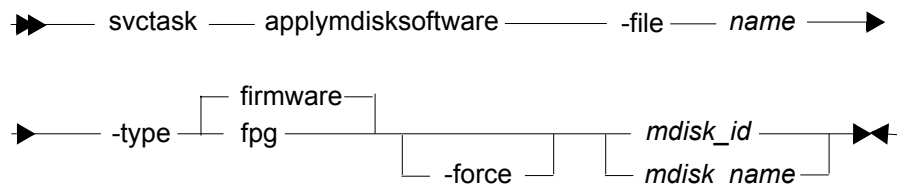
The following section has been corrected.

Page 135

Use the `applymdisksoftware` command to upgrade the firmware on a specified managed disk (MDisk). Only solid-state drives (SSDs) can be specified.

Syntax

```
svctask applymdisksoftware -file name  
  
-type firmware | fpga -force mdisk_id | mdisk_name
```



Parameters

-file *name*

(Required) Specifies the name of the firmware upgrade file. The file name must exist in the `/home/admin/upgrade/` directory.

-type *firmware* | *fpga*

(Required) Specifies the type of software that should be applied to the MDisk. The default is `firmware`. With the `fpga` option, the MDisk can remain offline for up to 20 minutes.

Attention: Only use the `fpga` option, which upgrades Field Programmable Gate Array (FPGA) firmware, under the direction of an IBM service representative.

-force

(Optional) Specifies that the upgrade should proceed even if VDIsks could go offline.

mdisk *id* | *name*

(Required) Specifies the ID or name of the MDisk to receive the firmware upgrade.

Description

The `applydisksoftware` command starts an upgrade of the firmware on a specified MDisk. When the command is run, the software image is loaded to the MDisk. If the upgrade could cause any VDIs to go offline, the **force** parameter is required. For example, a firmware update to a managed MDisk requires the **force** parameter.

An invocation example

```
svctask applydisksoftware -file SSDsoftware -type firmware mdisk1
```

The resulting output

No feedback

Chapter 16. Metro Mirror and Global Mirror commands

The following correction should be noted.

rmrelationship

The following section has been corrected.

Page 164

The `rmrelationship` command deletes an existing Metro Mirror or Global Mirror relationship.

Syntax

```
svctask rmrelationship rc_rel_id | rc_rel_name
```

Parameters

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 only deletes the logical relationship between the two virtual disks; it does not affect the virtual disks themselves.

If the relationship is disconnected at the time that the command is issued, the relationship is only deleted on the cluster where the command is being run. When the clusters reconnect, the relationship is automatically deleted on the other cluster. Alternatively, if the clusters are disconnected and if you still want to remove the relationship on both clusters, you can issue the `svctask rmrelationship` command independently on both of the clusters.

If you delete an inconsistent relationship, the secondary virtual disk becomes accessible even though it is still inconsistent. This is the one case in which Metro or Global Mirror does not inhibit access to inconsistent data.

An invocation example

```
svctask rmrelationship rcopy1
```

The resulting output

No feedback

Chapter 25. Service mode commands

The following correction should be noted.

cleardumps

The following section has been corrected.

Page 330

The **cleardumps** command cleans the various dump directories on the node that is in service mode.

Syntax

```
▶▶— svcservicemodetask — cleardumps —▶▶
▶ — -prefix — directory_or_file_filter —▶▶
```

Parameters

-prefix *directory_or_file_filter*

(Required) Specifies the directory, files, or both to be cleaned. If a directory is specified, with no file filter, all relevant dump or log files in that directory are cleaned. You can use the following directory arguments (filters):

- /dumps (cleans all files in all subdirectories)
- /dumps/cimon
- /dumps/configs
- /dumps/elogs
- /dumps/feature
- /dumps/iostats
- /dumps/iotrace
- /dumps/mdisk
- /home/admin/upgrade

In addition to the directory, you can specify a file filter. For example, if you specify `/dumps/elogs/*.txt`, all files in the **/dumps/elogs** directory that end in `.txt` are cleaned.

Note: 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.
- When you use a wildcard, enclose the filter entry within double quotation marks (""), as follows:
- `svcservicemodetask cleardumps -prefix "/dumps/elogs/*.txt"`

Description

This command deletes all the files that match *directory_or_file_filter* value for the node that is in service mode.

You can clean all the dumps directories by specifying **/dumps** as the directory value.

You can clean all the files in a single directory by specifying one of the directory values.

You can list the contents of these directories on the given node by using the **svcservicemodeinfo lsxxxxdumps** commands.

You can use this command to clean specific files in a given directory by specifying a directory or file name. You can use the wildcard (*) as part of your file name.

Note: To preserve the configuration and trace files, any files that match the following wildcard patterns are not cleaned:

- `*svc.config*`
- `*.trc`
- `*.trc.old`

An invocation example

```
svcservicemodetask cleardumps -prefix /dumps/configs
```

The resulting output

No feedback

Chapter 26. Service mode information commands

The following correction should be noted.

lsnodevpd

The following section has been corrected.

Page 341

The lsnodevpd command returns the vital product data (VPD) for the node.

Syntax

```
➤—— svcserviceinfo—— lsnodevpd ——➤  
└── -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.

Using 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 a colon character.

Description

This command returns the VPD for the node. Each field is reported on a new line. All fields are strings. The VPD is split into sections, each with a section heading. The number of fields in the section follows the 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 8F4, 8G4, and 8A4 nodes, the VPD displays the device serial number of the FC card as N/A.

An invocation example

```
svcinfo lsnodevdpd 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_scsi/ide_devices 2
BIOS_manufacturer IBM Corp.
BIOS_version -[D6E124AUS-1.01]-
BIOS_release_date 04/30/2009
system_manufacturer IBM
system_product IBM System x -[2145CF8]-
version 00
planar_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

```

