

IBM System Storage SAN Volume Controller



Command-Line Interface User's Guide - Errata

Version 4.1.0

Contents

About this guide	v
Who should use this guide	v
Emphasis	v
Syntax diagrams	v
Terminology	vii
CLI special characters	viii
Using wildcards in the SAN Volume Controller CLI.	viii
Data types and value ranges	ix
Last update	xv
Chapter 1. Cluster commands	1
chcluster.	1
Chapter 2. Virtual disk commands	5
expandvdisksize	5
rmvdisk	6
shrinkvdisksize	8
Chapter 3. Metro and Global Mirror commands	11
chpartnership.	11
chrconsistgrp	12
chrrelationship	12
mkpartnership	14
mkrconsistgrp	15
mkrrelationship.	16

rmpartnership	19
rmrcconsistgrp	20
rmrrelationship.	21
startrcconsistgrp.	22
startrcrelationship	24
stoprcconsistgrp	26
stoprcrelationship	27
switchrcconsistgrp	28
switchrcrelationship	29

Chapter 4. Information commands	31
lscluster	31
lsclustercandidate	34
lserrlogbyrrelationship	35
lslicense	36
lsrconsistgrp.	37
lsrrelationship	40
lsrrelationshipcandidate	42
lsrrelationshipprogress	44

Notices	45
Trademarks	46

Glossary	47
---------------------------	-----------

Index	57
------------------------	-----------

About this guide

This guide provides errata information that pertains to version 4 release 1 of the IBM System Storage SAN Volume Controller Command-Line Interface (CLI) User's Guide.

This guide contains the corrections and additions of the SAN Volume Controller Command-Line Interface (CLI) User's Guide on a per chapter basis.

Who should use this guide

This guide is intended for system administrators or others who install and use the SAN Volume Controller.

Before creating or modifying a new Mirror relationship between two virtual disks (VDisks), review this guide and note the changes that are marked with a vertical bar (|). This guide provides updated information to the Version 4 Release 1 SAN Volume Controller Command-Line Interface (CLI) User's Guide.

Emphasis

Different typefaces are used in this guide to show emphasis.


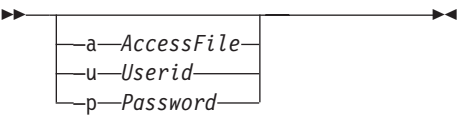

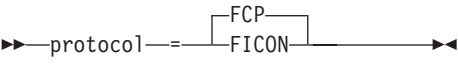
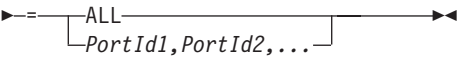
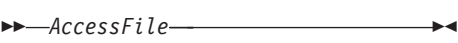
The following typefaces are used to show emphasis:

Boldface	Text in boldface represents menu items and command names.
<i>Italics</i>	Text in <i>italics</i> 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 cluster.
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.

This 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.

Element	Syntax	Description
Main path line	>>><>() ()	>>Begins on the left with double arrowheads (>>) and 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		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		Indicate the parameters or arguments you must specify for the command. Required keywords appear on the main path line. Mutually exclusive required keywords are stacked vertically.
Optional keywords		Indicate the parameters or arguments you can choose to specify for the command. Optional keywords appear below the main path line. Mutually exclusive optional keywords are stacked vertically.
Default value		Appears above the main path line.
Repeatable keyword or value		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		Represents the value you need to supply for a parameter or argument, such as a file name, user name, or password. Variables are in italics.

Element	Syntax	Description
Space separator	▶—u— — <i>Userid</i> — —p— — <i>Password</i> —▶	Adds a blank space on the main path line to separate keywords, parameters, arguments, or variables from each other.
Quotation mark delimiters	▶—d— —"— <i>ess</i> —="— <i>EssId</i> —▶ ▶—host—="—' <i>Host Name</i> ' —▶ ▶—profile—="— <i>ProfileName</i> —"—▶	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.
Equal-sign operator	▶—"— <i>ess</i> —="— <i>EssId</i> — —profile—▶ ▶—"— <i>ProfileName</i> —"—▶	Separates a name from its value in a name–value pair.
Syntax fragment	▶—Fragment Name—▶ Fragment name: —(— <i>fragment details</i> —)—	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.

Terminology

These are abbreviations that are most commonly used for the command-line interface operations.

The table below shows the abbreviations that are most commonly used for the command-line interface operations.

Name	Object type
Host	host
Virtual disk	vdisk
Managed disk	mdisk
Managed disk group	mdiskgrp
I/O group	iogrp
Node	node
Cluster	cluster
Controller	controller
FlashCopy [®] mapping	fcmap
FlashCopy consistency group	fcconsistgrp
Mirror relationship	rcrelationship

Name	Object type
Mirror consistency group	rcconsistgrp
Unsupported/unknown object	unknown

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] indicates that you can choose a, b, or nothing. Similarly, { a | b } indicates that you must choose either a or b.

Using wildcards in the SAN Volume Controller CLI

You can use wildcards in the SAN Volume Controller Command-Line Interface.

The SAN Volume Controller allows the use of the '*' 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 described below.

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. 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 SAN Volume Controller command untouched.

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

```
svctask clear.dumps -prefix '/dumps/*.txt' in single quotes, or
```

```
svctask clear.dumps -prefix /dumps/\*.txt using a backslash, or
```

```
svctask clear.dumps -prefix "/dumps/*.txt" in double quotes.
```

2. Running the command through SSH, for example from a host.

This is slightly more complicated since the host shell will process the command line before it is passed through SSH to the shell on the cluster. 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 cluster shell, then this will result in the wildcard being expanded in the cluster shell.

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

```
svctask clear.dumps "'/dumps/*.txt'" with single quotes inside double quotes, or
```

```
svctask clear.dumps '/dumps/\*.txt' using a backslash inside single quotes, or
```

```
svctask clear.dumps "'/dumps/*.txt'"
```

with double quotes inside single quotes.

Data types and value ranges

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

Note: If you do not specify a name when creating a new object, the cluster will assign a default name. This name is made from the object type as a prefix and the object ID as the suffix. For example, a new virtual disk (VDisk) is created with ID 5. This object will be given the default name of vdisk5. Because the system assigns these names, it will not allow you to create an object and call it vdiskx where *x* is the integer. This is because the cluster reserves these names (for example, object_type_prefix integer) for default.

Data types	Value ranges
filename_arg	<p>This is a (optionally fully qualified) file name. Maximum length is 169 characters. Valid characters consist of the following:</p> <ul style="list-style-type: none">• .• /• -• _• a - z• A - Z• 0 - 9 <p>The field may not contain two consecutive '.', or start with a '.', or end with a '.'.</p>

Data types	Value ranges
directory_or_file_filter	<p>Specifies a directory and or filename filter within the specified directory. Valid directory values consist of the following:</p> <ul style="list-style-type: none"> • /dumps • /dumps/audit • /dumps/configs • /dumps/elogs • /dumps/feature • /dumps/iostats • /dumps/iotrace • /dumps/software <p>The filename filter can be any valid filename with or without the wildcard '*'. The filename filter can be appended to the end of one of the above directory values. Maximum length is 128 characters. Valid characters consist of the following:</p> <ul style="list-style-type: none"> • * • . • / • - • _ • a - z • A - Z • 0 - 9 <p>The field may not contain two consecutive '.', or start with a '.', or end with a '.'.</p>
filename_prefix	<p>This is a prefix to be used when naming a file. Maximum length is 128 characters. Valid characters consist of the following:</p> <ul style="list-style-type: none"> • a - z • A - Z • 0 - 9 • - • _

Data types	Value ranges
name_arg	<p>Names can be specified or changed using the create and modify functions. The view commands allow you to see both the name and ID of an object.</p> <p>A string of 1-15 characters is allowed, composed of characters A-Z, a-z, 0-9, - and -.</p> <p>The first character of a name_arg must not be numeric. The first character of an object name can not be a - as the CLI will interpret it as being the next parameter.</p> <p>When creating a name for an object, the name may not consist of the object type followed only by an integer. The exception is Metro Mirror relationships which can be named anything as long as the names are unique across the two clusters. This naming convention is used by the system to generate default names. You can not use one of the following reserved words followed by an integer:</p> <ul style="list-style-type: none"> • cluster • controller • fccstgrp • fcmmap • host • io_grp • mdisk • mdiskgrp • node • rccstgrp • rcmap <p>The cluster name is set when the cluster is created.</p>
password	<p>This is a user defined password. A password must meet the following requirements:</p> <ul style="list-style-type: none"> • may use a - z, A - Z, 0 - 9 in any sequence • may use - (dash) but not as the first character • may use _ (underscore) • may contain a maximum of 15 characters
serial_number	<p>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.</p>

Data types	Value ranges
ip_address_arg	The decimal, dotted quad notation, standard rules.
dns_name	Dotted domain name for the subnet that the cluster is in. For example, <code>ibm.com[®]</code> .
hostname	<p>The hostname assigned to the cluster. This can be different from the cluster name and you can change the hostname at any time.</p> <p>A combination of the hostname and the <code>dns_name</code> that is used to access the cluster, for example:</p> <p><code>https://hostname.ibm.com/</code></p>
capacity_value	<p>A value with a range of 512 bytes up to 2 PetaBytes. The value can be expressed in multiples of 1 MB, ranging from 16 MB to 2 PetaBytes (PB).</p> <p>Note: The capacity can be specified as MB, KB, GB, or PB. When MB is used, the value is specified in multiples of 512 bytes. A capacity of 0 is valid for a striped/sequential vdisk. The smallest number of supported bytes is 512.</p>
delay_arg	Unassigned integer ranging from 1 to 65535 (minutes for battery test).
node_id	<p>Node IDs differ from other IDs as they are a unique node ID that is assigned when the node is initialized. Node IDs are expressed as 64-bit hexadecimal numbers. For example:</p> <p><code>1A2B30C67AFFE47B</code></p> <p>Node IDs, like other IDs, cannot be modified by user commands.</p>

Data types	Value ranges
xxx_id	<p>All objects are referred to by unique integer IDs that are assigned by the system when the objects are created. All IDs are represented internally as 32-bit integers. Node IDs are an exception.</p> <p>IDs in the following ranges are used to identify the various types of objects:</p> <ul style="list-style-type: none"> • node_id: 1 - 32 • mdisk_grp_id: 0 - 127 • io_grp_id: 0 - 3 (See Note.) • mdisk_id: 0 - 4095 • vdisk_id: 0 - 8191 • host_id: 0 - 127 • flash_const_grp_id: 0 - 255 • remote_const_grp_id: 0 - 255 • fcmapi_id: 0 - 4095 • rcrel_id: 0 - 8191 • controller_id: 0-63 <p>Note: io_group 4 exists but is only used in certain error recovery procedures.</p> <p>These IDs, like node IDs, cannot be modified by user commands.</p> <p>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. Therefore, wherever possible, object names should be used in preference to IDs when working with objects.</p>
xxx_list	A colon-delimited list of values of type <i>xxx</i> .
wwpn_arg	<p>The Fibre Channel World Wide Port Name (wwpn). This is expressed as a 64-bit hexadecimal number, for example:</p> <p>1A2B30C67AFFE47B</p> <p>These numbers must be composed of the characters 0 - 9, a - f, and A - F. A command will fail if you enter WWPN 0 in the command string.</p>
panel_name	A string of up to 6 characters that correspond to the number on the printed label below the APA display on the front panel of a node in the cluster.
sequence_number	32-bit unsigned integer, expressed in decimal.
csi_num_arg	32-bit unsigned integer, expressed in decimal.
percentage_arg	8-bit unsigned integer, expressed in decimal 0 to 100.

Data types	Value ranges
extent_arg	32-bit unsigned integer, expressed in decimal.
num_extents_arg	32-bit unsigned integer, expressed in decimal.
threads_arg	8-bit unsigned integer, expressed in decimal, valid values, 1, 2, 3, or 4.
velocity_arg	The fabric speed in Giga-bits per second. Valid values are 1 or 2.
timezone_arg	The ID as detailed in the output of the svcinfolstimezones command.
timeout_arg	The command timeout period. An integer from 0 to 600 (seconds).
stats_time_arg	The frequency at which statistics are gathered. 15 up to a max of 60 (minutes) in increments of 1 minute.
directory_arg	<p>Specifies a directory and or filename filter within the specified directory. Valid directory values are:</p> <ul style="list-style-type: none"> • /dumps • /dumps/audit • /dumps/configs • /dumps/elogs • /dumps/feature • /dumps/iostats • /dumps/iotrace • /home/admin/upgrade <p>The filename filter can be any valid filename with or without the wildcard '*'. The filename filter can be appended to the end of one of the above directory values.</p>
locale_arg	<p>The cluster locale setting. Valid values are 0 to 9.</p> <ul style="list-style-type: none"> • 0 US English (default) • 1 Chinese (simplified) • 2 Chinese (traditional) • 3 Japanese • 4 Korean • 5 French • 6 German • 7 Italian • 8 Spanish • 9 Portuguese (Brazilian)
key_arg	A user definable identifier for an SSH key. A string of up to 30 characters.
user_arg	Specifies the user, either admin or service.
copy_rate	A numeric value from 0 to 100.

Data types	Value ranges
copy_type	Specifies the mirror copy type, either metro or global.

The maximum number of values that can be entered into a colon separated list is 128. If more than 128 items are entered into a list an error is returned.

Last update

This guide was last updated July 5, 2006.

Chapter 1. Cluster commands

Cluster commands are used to monitor and modify clusters.

There are a number of cluster commands used for various tasks. A cluster is a pair of nodes that provide a single configuration and service interface.

chcluster

Use the **chcluster** command to modify the attributes of an existing cluster. You can enter this command any time after a cluster has been created.

Syntax

```
svctask -- chcluster [-clusterip cluster_ip_address]
                    [-serviceip service_ip_address] [-name cluster_name]
                    [-admpwd password] [-servicepwd password]
                    [-gw default_gateway] [-mask subnet_mask]
                    [-speed fabric_speed] [-alias id_alias]
                    [-icatip icat_console_ip_address]
                    [-gmlinktolerance link_tolerance]
                    [-gminterdelaysimulation inter_cluster_delay_simulation]
                    [-gmintradelaysimulation intra_cluster_delay_simulation]
```

Parameters

-clusterip *cluster_ip_address*

Changes the cluster IP address. After the cluster IP address is changed, you lose the open shell connection to the cluster. You must reconnect with the newly specified IP address.

-serviceip *service_ip_address*

Changes the service IP address. This address is the address that must be used if the node has to be started after it has been expelled from the cluster.

-name *cluster_name*

Changes the name of the cluster.

-admpwd *password*

Changes the administrator password. This argument can be specified with or without the password. If the argument is not followed by a password, you will be prompted for the password. When you type the password in response to the prompt, the password will not be displayed.

-servicepwd *password*

Changes the service user password. This argument can be specified with or without the password. If the argument is not followed by a password, you will be prompted for the password. When you type the password in response to the prompt, the password will not be displayed.

-gw *default_gateway*

Changes the default gateway IP address of the cluster.

-mask *subnet_mask*

Changes the subnet mask of the cluster.

-speed *fabric_speed*

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 will break I/O service to attached hosts. Before changing the fabric speed, stop I/O from active hosts and force such hosts to flush any cached data by dismounting volumes (for UNIX host types) or removing drive letters (for Windows host types). Some hosts may need to be rebooted to detect the new fabric speed.

The fabric speed setting applies only to the 4F2 and 8F2 model nodes in a cluster. The 8F4 nodes automatically negotiate the fabric speed on a per-port basis.

-alias *id_alias*

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.

-icatip *icat_console_ip_address*

Changes the IP address used by this cluster. The format of this IP address must be a dotted decimal notation together with the port (for example, 255.255.255.255:8080).

-gmlinktolerance *link_tolerance*

Specifies the response time of the primary VDisk in milliseconds. Specify values between 60 and 86400 in increments of 10. The default value is 60.

-gminterdelaysimulation *link_tolerance*

Specifies the number of milliseconds that I/O activity (intercluster copying to a secondary VDisk) is delayed. This permits you to test performance implications before deploying Global Mirror and obtaining a long distance link. Specify a value from 0 to 100 in 1 millisecond increments. The default value is 0. Use this argument to test each intercluster Global Mirror relationship separately.

-gmintradelaysimulation *link_tolerance*

Specifies the number of milliseconds that I/O activity (intracluster copying to a secondary VDisk) is delayed. This permits you to test performance implications before deploying Global Mirror and obtaining a long distance link. Specify a value from 0 to 100 in 1 millisecond increments. The default value is 0. Use this argument to test each intracluster Global Mirror relationship separately.

Description

This command modifies specific features of a cluster. The arguments are not mutually exclusive, so multiple features can be changed with a single command invocation.

If the cluster IP address is changed, the open command-line shell closes during the processing of the command. You must reconnect to the new IP address.

The service IP address is not used until a node is expelled from the cluster. If this node cannot rejoin the cluster, you can bring the node up in service mode. In this mode, the node can be accessed as a stand-alone node using the service IP address.

If you do not specify any of the options, the command does nothing.

Modifying a password: To change the administrator users password, issue the **svtask chcluster -admpwd <password>** command. To change the service users password, issue the **svtask chcluster -servicepwd <password>** command.

Note: If you do not wish the password to be displayed as you enter the command line then you can omit the new password. The command line tool will then prompt you to enter and confirm the password without the password being displayed.

Modifying an IP address: List the IP address of the cluster, by issuing the **svcinfo lscluster** command. Modify the IP address, by issuing the **svtask chcluster** command.

Possible failures

- CMMVC5786E The action failed because the cluster is not in a stable state.
- CMMVC5789E The cluster was not modified because the IP address, subnet mask, service address, SNMP address, or gateway address is not valid.

An invocation example

```
svctask chcluster -clusterip 217.12.3.11
```

The resulting output

No feedback

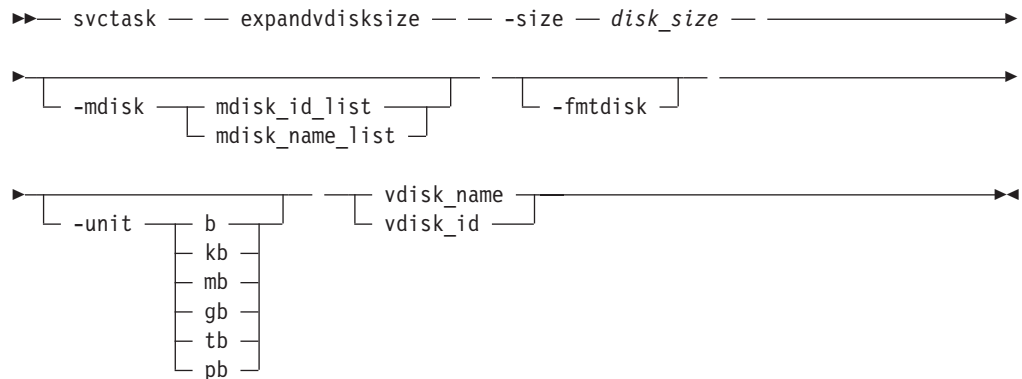
Chapter 2. Virtual disk commands

The following commands enable you to work with virtual disk options with the SAN Volume Controller.

expandvdisksize

The **expandvdisksize** command expands the size of a VDisk by a given capacity.

Syntax



Parameters

-size *disk_size*

Specifies the capacity by which the virtual disk is to be expanded. Disk size is used in conjunction with the unit's value. 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 will be reserved even if it is only partially used. The default capacity is in MB.

-mdisk *mdisk_id_list* | *mdisk_name_list*

Optionally specifies the list of one or more MDisks to be used as the stripe set. The extents that are used to expand the VDisk come from the specified list of MDisks. All MDisks in the list must be part of the same MDisk group.

-fmtdisk

Optionally specifies that the VDisk should be formatted before use. Using this parameter will format (all zeros) the new extents that have been added to the VDisk as a result of the expand.

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

Optionally specifies the data units to be used in conjunction with the capacity (-size).

vdisk_name | **vdisk_id**

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

Description

This command expands the capacity allocated to the particular virtual disk by the given amount. The default capacity is in MB.

When a VDisk is expanded, the policy may change. Its mode will become striped even if it was previously sequential. See the **svctask mkvdisk** command for details of the virtualization policies. This command will complete asynchronously if the **fmtdisk** argument is selected.

Possible failures

- CMMVC5786E The action failed because the cluster is not in a stable state.
- CMMVC5808E The action failed because the managed disk (MDisk) does not exist.
- CMMVC5835E The virtual disk (VDisk) was not expanded because an object that was specified in the command does not exist.
- CMMVC5837E The action failed because the virtual disk (VDisk) is part of a FlashCopy mapping.
- CMMVC5838E The action failed because the virtual disk (VDisk) is part of a Metro Mirror mapping.
- CMMVC5848E The action failed because the virtual disk (VDisk) does not exist or it is being deleted.
- CMMVC5853E The action failed because there was a problem with the group.
- CMMVC5856E The action failed because the virtual disk (VDisk) does not belong to the specified managed disk (MDisk) group.
- CMMVC5857E The action failed because the managed disk (MDisk) does not exist or it is not a member of the managed disk (MDisk) group.
- CMMVC5858E The action failed because the virtual disk (VDisk) is in the wrong mode, the managed disk (MDisk) is in the wrong mode, or both are in the wrong mode.
- CMMVC5860E The action failed because there were not enough extents in the managed disk (MDisk) group.
- CMMVC5861E The action failed because there were not enough extents on the managed disk (MDisk).
- CMMVC5862E The action failed because the virtual disk (VDisk) is being formatted.
- CMMVC5998W The virtualized storage capacity exceeds the amount that you are licensed to use.
- CMMVC6036E An invalid action was requested

An invocation example

```
svctask expandvdisksize -size 2048 -unit b -mdisk  
mdisk0:mdisk1 -fmtdisk vdisk1
```

The resulting output

No feedback

rmvdisk

The **rmvdisk** command deletes a virtual disk. The command will fail if there are mappings between the virtual disk and hosts and you do not specify the force flag.

Syntax

```
svctask — — rmvdisk — [ -force ] [ vdisk_id | vdisk_name ]
```

Parameters

-force

Optionally forces the deletion. This argument deletes any host-to-VDisk mappings and any FlashCopy mappings that exist for this VDisk.

vdisk_id | vdisk_name

Specifies the name of the virtual disk to delete, either by ID or by name.

Description

This command deletes an existing managed mode virtual disk or an existing image mode virtual disk. The extents that made up this virtual disk are returned to the pool of free extents that are available on the managed disk group, if the vdisk is in managed mode.

Attention: Any data that was on the virtual disk is lost. Care should be taken when executing this command to ensure that the virtual disk (and any data that resides on it) is no longer required.

If any mappings still exist between this virtual disk and hosts, the deletion will fail unless the force flag is specified. When the force flag is specified, any mappings that remain are deleted and then the virtual disk is deleted.

Deleting a managed mode virtual disk

When you use this command to delete a managed mode virtual disk, all the data on the virtual disk is deleted. The extents that make up the virtual disk are returned to the pool of free extents that are available in the managed disk group.

If FlashCopy mappings or host mappings exist for the virtual disk, then the deletion will fail. You can use the force flag to force the deletion. If you use the force flag, mappings are deleted, and then the virtual disk is deleted.

If the virtual disk is in the process of migrating to an image mode virtual disk (using the **svctask migratetoimage** command), then the deletion will fail unless the force flag is used. In this case, the migration is halted and then the virtual disk is deleted. Care should be taken when executing this command to ensure that the virtual disk (and any data that resides on it) is no longer required.

Deleting an image mode virtual disk

When you use this command to delete an image mode virtual disk, the data on the controller logical unit will be consistent with the data that had been available on the image mode virtual disk before it was deleted. In other words, the fast write data will be moved to the controller logical unit. If the force flag is used, then the data is not moved to the controller logical unit.

If there are any virtual medium errors on the virtual disk, then the command will fail. You can force the deletion with the force flag, however this option can cause data integrity issues.

Note: A virtual medium error occurs when you copy from one disk (the source) to another (the target). If you read the source, you should detect that there is a medium error. At that moment you have to have two identical copies of data and you then have to simulate a medium error on the target disk. You can simulate that 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 virtual disk, then the deletion will fail. You can use the force flag to force the deletion. If you use the force flag, mappings are deleted, and then the virtual disk is deleted. If there is any un-dstaged data in the fast write cache for this virtual disk then the deletion of the virtual disk will fail. When the force flag is specified, any un-dstaged data in the fast write cache will be deleted. Deleting an image mode virtual disk causes the managed disk associated with the virtual disk to be ejected from the managed disk group. The mode of the managed disk will be returned to “unmanaged.”

Possible failures

- CMMVC5786E The action failed because the cluster is not in a stable state.
- CMMVC5807E The action failed because the managed disk (MDisk) cannot be changed to the specified mode.
- CMMVC5840E The virtual disk (VDisk) was not deleted because it is mapped to a host or because it is part of a FlashCopy or Metro Mirror mapping.
- CMMVC5841E The virtual disk (VDisk) was not deleted because it does not exist.
- CMMVC5848E The action failed because the virtual disk (VDisk) does not exist or it is being deleted.
- CMMVC5858E The action failed because the virtual disk (VDisk) is in the wrong mode, the managed disk (MDisk) is in the wrong mode, or both are in the wrong mode.
- CMMVC5862E The action failed because the virtual disk (VDisk) is being formatted.

An invocation example

```
svctask rmvdisk -force vdisk5
```

The resulting output

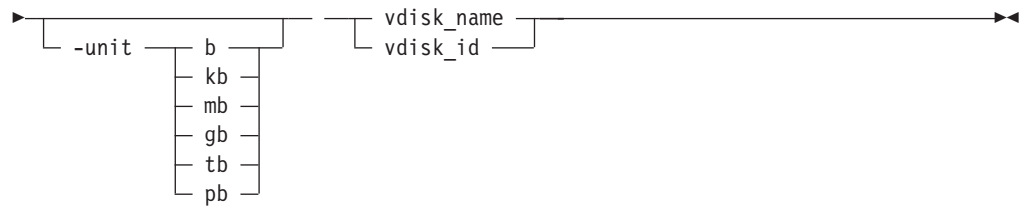
No feedback

shrinkvdisksize

You can use the **shrinkvdisksize** command to shrink a VDisk by a given capacity.

Syntax

```
▶▶ svctask — — shrinkvdisksize — — -size — disk_size — —————▶
```

Parameters

-size *disk_size*

The command line will reduce the size by the specified amount.

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

Optionally specifies the data units to be used in conjunction with the capacity (-size).

vdisk_name | vdisk_id

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

Description

This command reduces the capacity that is allocated to the particular virtual disk by the given amount. 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 will be reserved even if it is only partially used. The default capacity is in MB.

VDisks can be reduced in size should it be required. However, if the VDisk contains data that is being used, **under no circumstances should you attempt to shrink a VDisk without first backing up your data**. The cluster arbitrarily reduces the capacity of the VDisk by removing a partial, one or more extents from those allocated to the VDisk. You cannot control which extents are removed and so you cannot assume that it is unused space that is removed.

Attention: This feature should *only* be used to make a target or auxiliary VDisk the same size as the source or master VDisk when creating FlashCopy mappings or Metro Mirror relationships. You should also ensure that the target VDisk is not mapped to any hosts prior to performing this operation.

Attention: If the virtual disk contains data, you should not shrink the disk.

Note: Some operating systems or file systems use what they consider to be the outer edge of the disk for performance reasons. This command is provided to shrink FlashCopy target virtual disks to the same capacity as the source.

Validate that the VDisk is not mapped to any host objects. If the VDisk is mapped, data is displayed. You can determine the exact capacity of the source or master VDisk by issuing the **svcinfolsvdisk -bytes <vdiskname>** command. Shrink the VDisk by the required amount by issuing the **svctask shrinkvdisksize -size <capacitytoshrinkby> -unit <unitsforreduction> <vdiskname/ID>** command.

Possible failures

- CMMVC5786E The action failed because the cluster is not in a stable state.
- CMMVC5836E The virtual disk (VDisk) was not shrunk because it is locked.

- CMMVC5837E The action failed because the virtual disk (VDisk) is part of a FlashCopy mapping.
- CMMVC5838E The action failed because the virtual disk (VDisk) is part of a Metro Mirror mapping.
- CMMVC5839E The virtual disk (VDisk) was not shrunk because an entity that was specified in the command does not exist.
- CMMVC5848E The action failed because the virtual disk (VDisk) does not exist or it is being deleted.
- CMMVC5862E The action failed because the virtual disk (VDisk) is being formatted.
- CMMVC6010E Unable to complete the command as there are insufficient free extents.

An invocation example

```
svctask shrinkvdisksize -size 2048 -unit b vdisk1
```

The resulting output

No feedback

Chapter 3. Metro and Global Mirror commands

The following Copy Services commands enable you to work with the Global Mirror and Metro Mirror services provided by the SAN Volume Controller.

chpartnership

Use the **chpartnership** command to specify the bandwidth available for background copy in a cluster partnership that has been created for Mirror purposes.

Syntax

```
svctask -- chpartnership -- -bandwidth -- bandwidth_in_mbps --  
└─ remote_cluster_id ─┘  
└─ remote_cluster_name ─┘
```

Parameters

-bandwidth *bandwidth_in_mbps*

Specifies the new bandwidth in megabytes per second (MBps). This argument might be set to a value that is greater than the intercluster links can sustain. If so, the actual copy rate defaults to what is available on the link. The default bandwidth is 50.

remote_cluster_id | **remote_cluster_name**

Specifies the cluster ID or name of the remote cluster. The intracluster bandwidth cannot be modified so if you enter the local cluster name or ID, an error will occur.

Description

This command modifies the bandwidth of the partnership between the local cluster and the remote cluster that is specified in the command. This affects the bandwidth available for background copy in Mirror relationships, in the direction from the local to the remote cluster. To modify the background copy bandwidth in the other direction (remote cluster → local cluster), it is necessary to issue the corresponding **chpartnership** command to the remote cluster.

Possible failures

- CMMVC5786E The action failed because the cluster is not in a stable state.
- CMMVC5927E The action failed because the cluster ID is not valid.

An invocation example

```
svctask chpartnership -bandwidth 20 cluster1
```

The resulting output

No feedback

chrconsistgrp

Use the **chrconsistgrp** command to modify the name of an existing Mirror consistency group.

Syntax

```
svctask -- chrconsistgrp -- -name -- new_name_arg --  
└── rc_consist_group_name ────────────────────────────────────┘  
└── rc_consist_group_id ────────────────────────────────────┘
```

Parameters

-name *new_name_arg*

Specifies the new name to assign to the consistency group.

rc_consist_group_name | **rc_consist_group_id**

Specifies the ID or existing name of the consistency group to modify.

Description

This command changes the name of the specified consistency group.

Possible failures

- CMMVC5786E The action failed because the cluster is not in a stable state.
- CMMVC5937E The action failed because an entity that was specified in the command does not exist.

An invocation example

Change the name of the Mirror consistency group called rc_testgrp to rctestone.

```
svctask chrconsistgrp -name rctestone rc_testgrp
```

The resulting output

No feedback

chrrelationship

The **chrrelationship** command enables you to modify certain attributes of an existing relationship such as add a relationship to a consistency group, remove a relationship from a consistency group, and change the name of the relationship.

Syntax

```
svctask -- chrrelationship --  
└── -name -- new_name_arg ────────────────────────────────────┘  
└── -force ────────────────────────────────────────────────────┘  
└── -consistgrp ───────────┬── consist_group_id ───────────┘  
└── consist_group_name ────────────────────────────────────┘  
└── rc_rel_id ─────────────────────────────────────────────────┘  
└── rc_rel_name ─────────────────────────────────────────────────┘
```

Parameters

-name *new_name_arg*

Optionally specifies a new label to assign to the relationship

-consistgrp *consist_group_id* | *consist_group_name*

Optionally specifies a new consistency group to assign the relationship to.

Only relationships of the same copy type (Metro Mirror or Global Mirror) can be assigned to the same consistency group.

-force

Optionally specifies the force flag which will remove the relationship from a consistency group making the relationship a standalone relationship.

rc_rel_name | **rc_rel_id**

Specifies the ID or name of the relationship.

Note: The **-name**, **-consistgrp** and **-force** are mutually exclusive parameters. That is, only one of these parameters can be specified per command line.

Description

This command can modify the specified attributes of the relationship supplied. Only one attribute can be modified at a time. That is, all three optional flags are mutually exclusive. In addition to changing the name of a consistency group, this command can be used for the following purposes.

- **Add a relationship to a group.** A standalone relationship can be added to a consistency group by specifying the **-consistgrp** parameter and the name or ID of the consistency group. The relationship and consistency group must both be connected when the command is issued, and both must have the same:
 - Master cluster
 - Auxiliary cluster
 - State (unless the group is empty)
 - Primary (unless the group is empty)
 - Type (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), and type (Metro or Global Mirror) 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 only belong to one consistency group.

- **Remove a relationship from a group.** A relationship can be removed from a consistency group by simply specifying the **-force** flag and the name or ID of the relationship. You do not have to specify or confirm the name of the consistency group, so it is recommended that you verify which group the relationship belongs to before issuing this command.

This form of the modify relationship command will succeed in the connected or disconnected states. If the clusters are disconnected, then the relationship will only be removed from the consistency group on the local cluster, at the time the command is issued. When the clusters are reconnected, the relationship will automatically be removed from the consistency group on the other cluster. Alternatively, you can issue an explicit modify (**chrrelationship**) command to remove the relationship from the group on the other cluster while it is still disconnected.

Note: If you remove all relationships from the group, the relationship type is reset to null. Then, when you add a relationship to the empty group, the group again takes on the same type as the relationship.

- **Move a relationship from one group to another** To move a relationship between two consistency groups you must issue the `chrrelationship` command twice. First use the `-force` flag to remove the relationship from its current group, then use the `-consistgrp` parameter and the name of the new consistency group it is to be added to.

Possible failures

- CMMVC5786E The action failed because the cluster is not in a stable state.
- CMMVC5935E The action failed because an entity that was specified in the command does not exist.
- CMMVC6234E The copy type differs from other copies already in the consistency group.

An invocation example

Change the name of the relationship `rccopy1` to `testrel`

```
svctask chrrelationship -name testrel rccopy1
```

Add relationship `rccopy2` to group called `newgroup`.

```
svctask chrrelationship -consistgrp newgroup rccopy2
```

Remove relationship `rccopy3` from whichever consistency group it is a member of.

```
svctask chrrelationship -force rccopy3
```

The resulting output

No feedback

There is no feedback in any of these cases.

mkpartnership

You can use the **mkpartnership** command to establish a one-way Mirror partnership between the local cluster and a remote cluster.

Syntax

To establish a fully functional Mirror partnership, you must issue this command to both clusters. This step is a prerequisite to creating Mirror relationships between VDisks on the clusters.

```
svctask -- mkpartnership -- -bandwidth bandwidth_in_mbps remote_cluster_id  
remote_cluster_name
```

Parameters

-bandwidth *bandwidth_in_mbps*

Optionally specifies the bandwidth to be used by the background copy process between the clusters. It can be used to throttle the bandwidth used by Metro or

Global Mirror for the initial background copy process. The bandwidth defaults to 50 MBps (megabytes per second) if you do not specify it. The bandwidth should be set to a value that is less than or equal to the bandwidth that can be sustained by the intercluster link. If the parameter is set to a higher value than the link can sustain, the background copy process will simply use the actual available bandwidth. Refer to the *IBM System Storage SAN Volume Controller: Configuration Guide* for more information about the effect background copy bandwidth has on foreground I/O latency.

remote_cluster_id | remote_cluster_name

Specifies the cluster ID or name of the remote cluster. You can use the **svcinfolsclostercandidate** command to list the remote clusters that are available. If two or more remote clusters have the same name and the name is included in this command, the command will fail and ask for the ID of the cluster instead of the name.

Description

This command creates a one-way partnership between the local cluster and the remote cluster that you specify in the command. To create a two-way partnership, the equivalent **svctask mkpartnership** command must be issued from the other cluster.

Intercluster Mirror relationships can be created between primary VDisks in the local cluster and auxiliary VDisks in the remote cluster. Intracluster relationships can be created between VDisks that reside in a local cluster. The VDisks must belong to the same I/O group within the cluster.

Possible failures

- CMMVC5786E The action failed because the cluster is not in a stable state.
- CMMVC5925E The remote cluster partnership was not created because it already exists.
- CMMVC5926E The remote cluster partnership was not created because there are too many partnerships.
- CMMVC5927E The action failed because the cluster ID is not valid.
- CMMVC5928E The action failed because the cluster name specified is a duplicate of another cluster.

An invocation example

```
svctask mkpartnership -bandwidth 20 cluster1
```

The resulting output

No feedback

mkrcconsistgrp

The **mkrcconsistgrp** command creates a new, empty Mirror consistency group.

Syntax

```
svctask mkrcconsistgrp -name new_name_arg
```



Parameters

-name *new_name_arg*

Optionally specifies a name for the new consistency group.

-cluster *cluster_id* | *cluster_name*

To create an intercluster consistency group, enter the ID or name of the remote cluster. If **-cluster** is not specified, then an intracluster consistency group is created on the local cluster only.

Description

This command creates a new consistency group. The ID of the new group is returned. The name must be unique across all consistency groups known to the clusters owning this consistency group. If the consistency group involves two clusters, the clusters must be in communication throughout the create process.

The new consistency group does not contain any relationships and will be in the empty state. Mirror relationships can be added to the group using the **svctask chrelationship** command.

Possible failures

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

An invocation example

```
svctask mkrcconsistgrp -name rc_testgrp
```

The resulting output

```
RC Consistency Group, id [255], successfully created
```

mkrcrelationship

The **mkrcrelationship** command creates a new Mirror relationship with virtual disks (VDisks) in the same cluster (intracluster relationship) or in two different clusters (intercluster relationship).

Syntax

Parameters

-master *master_vdisk_id* | *master_vdisk_name*

Specifies the ID or name of the master virtual disk.

-aux *aux_vdisk_id* | *aux_vdisk_name*

Specifies the ID or name of the auxiliary virtual disk.

-cluster *cluster_id* | *cluster_name*

Specifies the ID or name of the remote cluster.

If you are creating an intracluster relationship, enter the ID of the local cluster. The VDIs in the relationship must belong to the same I/O group within the cluster.

If you are creating an intercluster relationship, enter the ID of the remote cluster. To create a relationship in two different clusters, the clusters must be connected at the time that the **svctask mkrrelationship** command is received.

-name *new_name_id*

Optionally specifies a label to assign to the relationship.

-consistgrp *consist_group_id* | *consist_group_name*

Optionally specifies a consistency group that this relationship will join. If you do not supply the **-consistgrp** argument, the relationship will be a standalone relationship that can be started, stopped, and switched on its own.

Note: Metro and Global Mirror 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. Subsequently, only relationships of that type can be added to the consistency group.

-sync

Optionally specifies the synchronized, or create consistency flag. Use this argument to indicate that the secondary (auxiliary) virtual disk is already synchronized with the primary (master) virtual disk. The initial background synchronization is skipped.

-global

Optionally specifies that a new Global Mirror relationship is to be created. If you do not supply the **-global** argument, a Metro Mirror relationship is created.

Description

This command creates a new Mirror relationship. A Mirror relationship defines the relationship between two virtual disks (VDIs): a master VDisk and an auxiliary VDisk. This relationship persists until it is deleted. The auxiliary virtual disk must be identical in size to the master virtual disk or the command will fail, and if both VDIs are in the same cluster they must both be in the same I/O group. The master and auxiliary cannot be in an existing relationship. Neither disk can be the target of a FlashCopy mapping. This command returns the new relationship (*relationship_id*) when successful.

Mirror relationships use one of the following copy types:

A Metro Mirror copy ensures that updates are committed to both the primary and secondary VDIs before sending confirmation of I/O completion to the host application. This ensures that the secondary VDisk is synchronized with the primary VDisk in the event that 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 VDisk. If a
| failover operation is performed, the host application must recover and apply
| any updates that were not committed to the secondary VDisk.

You can optionally give the relationship a name. The name must be a unique relationship name across both clusters.

| The relationship can optionally be assigned to a Mirror consistency group. A
| consistency group is used to ensure that a number of relationships are managed so
| that, in the event of a disconnection of the relationships, the data in all
| relationships within the group is in a consistent state. This can be important in,
| say, a database application where data files and log files are held on separate
| VDisks, and consequently are being managed by separate relationships. In the
| event of a disaster, the primary and secondary sites may become disconnected. If
| the relationships associated with the VDisks are not in a consistency group, then as
| the disconnection happens, and the Mirror relationships stop copying data from
| the primary to the secondary site, there is no assurance that updates to the two
| separate secondary VDisks will stop in a consistent manner.

| For proper database operation, though, it is important that updates to the log files
| and the database data are made in a consistent and orderly fashion. It is thus
| crucial in this example that the logfile VDisk and the data VDisk at the secondary
| site are in a consistent state. This can be achieved by putting the relationships
| associated with these VDisks into a consistency group. A Mirror copy then
| ensures that updates to both VDisks at the secondary site are consistent with the
| updates that have been made at the primary site.

| If you specify a consistency group, both the group and the relationship must have
| been created using the same master cluster and the same auxiliary cluster. The
| relationship must not be a part of another consistency group. If the consistency
| group is *empty*, then it will acquire the type of the relationship that is added to it.
| Subsequently, each relationship that you add to the consistency group must be of
| the same type.

If the consistency group is *not empty*, then the consistency group and the relationship must be in the same state. If the consistency group is *empty*, then it will acquire the state of the first relationship that is added to it. If the state has a copy direction assigned, then the direction of the consistency group and the relationship must match that direction.

If you do not specify a consistency group, a standalone relationship is created.

If you specify the `-sync` argument, it is taken as an assurance that the master and auxiliary virtual disks 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 write operation takes place to either virtual disk before you issue the **svctask mkrrelationship** command.

| If you specify the `-global` argument, a Global Mirror relationship is created.
| Otherwise, a Metro Mirror relationship is created.

Possible failures

Note: If you receive an error for this command that indicates that the licensed virtualization capacity has been exceeded, then the command was still effective. However, the return code will indicate the license violation.

- CMMVC5786E The action failed because the cluster is not in a stable state.
- CMMVC5930E The Metro Mirror relationship was not created because an entity that was specified in the command does not exist.
- CMMVC5931E The Metro Mirror relationship was not created because the master or auxiliary virtual disk (VDisk) is locked.
- CMMVC5932E The Metro Mirror relationship was not created because the master or auxiliary virtual disk (VDisk) is a member of a FlashCopy mapping.
- CMMVC5933E The Metro Mirror relationship was not created because the master or auxiliary virtual disk (VDisk) is in the recovery I/O group.
- CMMVC5934E The specified relationship is not valid.
- CMMVC5965E The virtual disks (VDisks) are in different I/O groups on the local cluster.
- CMMVC6024E The auxiliary VDisk entered is invalid.
- CMMVC6034E The action failed because the maximum number of objects has been reached.

An invocation example

```
svctask mkrcrelationship -master vdisk1 -aux vdisk2 -name rccopy1  
-cluster 0000020063432AFD
```

The resulting output

```
RC Relationship, id [28], successfully created
```

rmpartnership

The **rmpartnership** command removes a Mirror partnership.

Syntax

Because the partnership exists on both clusters, it is necessary to run this command on both clusters to remove both sides of the partnership. If the command is run on only one cluster, then the Mirror partnership will enter a partially configured state and Mirror activity will cease as the relationships become disconnected.

```
►► svctask — rmpartnership — [ remote_cluster_id | remote_cluster_name ] ►►
```

Parameters

remote_cluster_id | **remote_cluster_name**

Specifies the cluster ID or name of the remote cluster.

Description

This command deletes the partnership between the local cluster and the remote cluster specified in the command.

Possible failures

- CMMVC5786E The action failed because the cluster is not in a stable state.
- CMMVC5927E The action failed because the cluster ID is not valid.
- CMMVC5928E The action failed because the cluster name is a duplicate of another cluster.
- CMMVC5929E The Metro Mirror partnership was not deleted because it has already been deleted.

An invocation example

```
svctask rmpartnership cluster1
```

The resulting output

No feedback

rmrcconsistgrp

The **rmrcconsistgrp** command deletes an existing Mirror consistency group.

Syntax

```
svctask - - rmrcconsistgrp - [-force] rc_consist_group_id | rc_consist_group_name
```

Parameters

-force

If the group contains any relationships and you do not specify the force flag, the command will fail. If one or more relationship belongs to the group and you do not specify the force flag, the deletion fails. If you specify the force flag, any relationships belonging to the group are removed from the group before it is deleted. The relationships themselves are not deleted; they become stand-alone relationships.

rc_consist_group_id | rc_consist_group_name

Specifies the ID or 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, then the consistency group is only deleted on the cluster on which the command is being run. When the clusters reconnect, then 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 **svctask rmrcconsistgrp** command separately on both of the clusters.

If the consistency group is not empty, the **-force** parameter is required to delete the group. This will remove the relationships from the consistency group before the group is deleted. These relationships then become standalone relationships. The state of these relationships is not changed by the action of removing them from the consistency group.

Possible failures

- CMMVC5786E The action failed because the cluster is not in a stable state.
- CMMVC5937E The action failed because an entity that was specified in the command does not exist.
- CMMVC5938E The Metro Mirror consistency group was not deleted because the consistency group contains relationships. To delete the consistency group, the force option is required.

An invocation example

```
svctask rmrconsistgrp rctestone
```

The resulting output

```
No feedback
```

rmrrelationship

The **rmrrelationship** command deletes an existing Mirror relationship.

Syntax

```
svctask -- rmrrelationship -- rc_rel_id rc_rel_name
```

Parameters

rc_rel_id | **rc_rel_name**

Specifies the ID or name of the relationship. A relationship cannot be deleted if it is part of a consistency group.

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, then the relationship is only deleted on the cluster on which the command is being run. When the clusters reconnect, then the relationship is automatically deleted on the other cluster. Alternatively, if the clusters are disconnected, and you still want to remove the relationship on both clusters, you can issue the **svctask rmrrelationship** command independently on both of the clusters.

A relationship cannot be deleted if it is part of a consistency group. You must first remove the relationship from the consistency group using the **svctask chrrelationship -force** command.

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

Possible failures

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

- CMMVC5935E The action failed because an entity that was specified in the command does not exist.

An invocation example

```
svctask rmrcrelationship rccopy1
```

The resulting output

No feedback

startrcconsistgrp

You can use the **startrcconsistgrp** command to start the Mirror consistency group copy process, set the direction of copy if undefined, and optionally mark the secondary VDisks of the consistency group as clean.

Syntax

```
svctask -- startrcconsistgrp -- [-primary master | aux] rc_consist_group_id [-force] [-clean] rc_consist_group_name
```

Parameters

-primary *master* | *aux*

This parameter specifies the copy direction by defining whether the master or auxiliary will become the primary (source). This parameter is required when the primary is undefined, for example if the consistency group is in the idling state. The primary (direction) argument specifies which disk is the primary, that is, the source disk.

-force

Optionally specifies the force parameter. This argument permits the copy operation to resume even if it might lead to a temporary loss of consistency while synchronization occurs.

-clean

Optionally specifies the clean parameter. This marks the secondary VDisk as clean for each of the relationships belonging to the group.

rc_consist_group_id | **rc_consist_group_name**

Specifies the ID or name of the consistency group to start.

Description

This command starts a Mirror consistency group.

This command can only be issued 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 some I/O error.

If the resumption of the copy process will lead to a period when the relationship is not consistent, then you must specify the force flag when restarting the

relationship. This situation could arise if, say, the relationship had been stopped, and then further writes had been performed on the original primary of the relationship. The use of the force flag here is a reminder that the data on the secondary will not be useful for Disaster Recovery purposes while it is in an inconsistent state.

In the idling state, you must provide the primary argument. In other connected states, you can provide the primary argument, but it must match the existing setting.

The `-force` flag is required if consistency would be lost by starting a copy operation. This would occur, if write operations on either primary or secondary vdisks have taken place since the `ConsistentStopped` or idling state was entered. If the command is issued without the `-force` flag in such circumstances, the command will fail. In general, the `-force` flag will be required if the group is in one of the following states:

- `ConsistentStopped` but not synchronized (`sync=out_of_sync`)
- `Idling` but not synchronized

The `-force` flag is not required if the group is in one of the following states:

- `InconsistentStopped`
- `InconsistentCopying`
- `ConsistentSynchronized`

However, the command will not fail if you do specify the `-force` flag.

The `clean` flag is used when a Mirror group is started and the secondary VDisks in this group are assumed to be clean. Clean in this sense means that any changes that have been made at the secondary are ignored and only changes made at the primary are considered when synchronizing primary and secondary. This flag could be used in the following scenario:

1. A consistency group is created with the `synchronized` flag. At this point it does not matter if the primary and secondary contain the same data, even though the use of the `synchronized` flag implies that this is true.
2. A `stoprconsistgrp` command is issued with the `-allow access` flag. This permits access to the secondary. Change recording begins at the primary.
3. An image of the primary is captured and loaded on to the secondary. It is permissible to allow updates to the primary during the image copy as this image need only be a fuzzy image of the primary.
4. A `starttrconsistgrp` command with `primary = master`, the `force` flag and the `clean` flag is issued. This causes the auxiliary to be marked clean, and changes on the master that have occurred since the consistency group was stopped are copied to the auxiliary.
5. Once the background copy has completed, relationships in the group will be consistent and synchronized.

Possible failures

- CMMVC5786E The action failed because the cluster is not in a stable state.
- CMMVC5936E The action failed because an entity that was specified in the command does not exist.

An invocation example

```
svctask starttrconsistgrp rccopy1
```

The resulting output

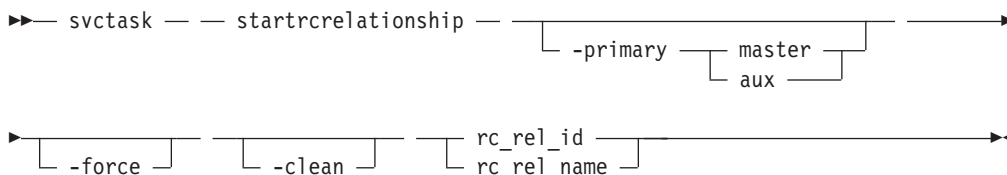
No feedback

startrelationship

Use the **startrelationship** command to start the Mirror relationship copy process, set the direction of copy if undefined, and optionally mark the secondary VDisk of the relationship as clean.

Syntax

```
svctask -- startrelationship [-primary master | aux] [-force] [-clean] [rc_rel_id | rc_rel_name]
```



Parameters

-primary *master* | *aux*

Specifies the copy direction by defining whether the master or auxiliary will become the primary (source). This parameter is required when the primary is undefined, for example, if the relationship is in the idling state.

-force

Optionally specifies the force parameter. This argument permits the copy operation to resume even if it might lead to a loss of consistency.

-clean

Optionally specifies the clean parameter. The clean flag marks the secondary virtual disk as clean.

rc_rel_id | **rc_rel_name**

Specifies the ID or name of the relationship that you want to start in a stand-alone relationship only.

Description

This command is used to start a standalone relationship. The command will fail if it is used to attempt to start a relationship that is part of a consistency group.

This command can only be issued 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.

If the resumption of the copy process will lead to a period when the relationship is not consistent, then you must specify the force flag when restarting the relationship. This situation could arise if, say, the relationship had been stopped, and then further writes had been performed on the original primary of the relationship. The use of the force flag here is a reminder that the data on the secondary will not be useful for disaster recovery purposes while it is in an inconsistent state.

In the idling state, you must provide the primary argument. In other connected states, you can provide the primary argument, but it must match the existing setting.

The `-force` flag is required if consistency would be lost by starting a copy operation. This would occur, if write operations on either primary or secondary vdisks have taken place since the `ConsistentStopped` or idling state was entered. If the command is issued without the `-force` flag in such circumstances, the command will fail. In general, the `-force` flag will be required if the relationship is in one of the following states:

- `ConsistentStopped` but not synchronized
- Idling but not synchronized

The `-force` flag is not required if the relationship is in one of the following states:

- `InconsistentStopped`
- `InconsistentCopying`
- `ConsistentSynchronized`

However, the command will not fail if you do specify the `-force` flag.

The `clean` flag is used when a Mirror relationship is started and the secondary VDisk in this relationship is assumed to be clean. Clean in this sense means that any changes that have been made at the secondary are ignored and only changes made at the primary are considered when synchronizing primary and secondary. This flag could be used in the following scenario:

1. A relationship is created with the `synchronized` flag. (At this point it does not matter if the primary and secondary contain the same data, even though the use of the `synchronized` flag implies that this is true).
2. A `svctask stopprrelationship` command is issued with the `-allow access` flag. This permits access to the secondary. Change recording begins at the primary.
3. An image of the primary is captured and loaded on to the secondary. It is permissible to allow updates to the primary during the image copy as this image need only be a 'fuzzy' image of the primary.
4. A `svctask starttrrelationship` command with `primary = master`, the `force` flag and the `clean` flag is issued. This causes the auxiliary to be marked clean, and changes on the master that have occurred since the relationship was stopped are copied to the auxiliary.
5. Once the background copy has completed, the relationship will be consistent and synchronized.

Possible failures

- CMMVC5786E The action failed because the cluster is not in a stable state.
- CMMVC5936E The action failed because an entity that was specified in the command does not exist.

An invocation example

```
svctask starttrrelationship rccopy1
```

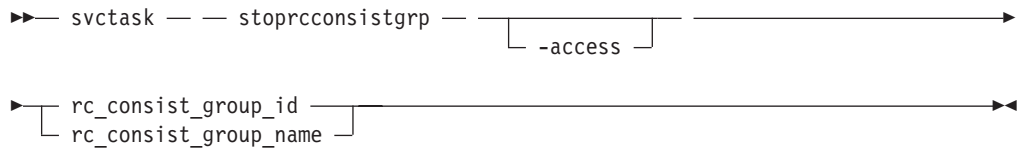
The resulting output

```
No feedback
```

stoprconsistgrp

You can use the **stoprconsistgrp** command to stop the copy process in a Mirror consistency group. It can also be used to enable write access to the secondary VDisks in the group if the group is in a consistent state.

Syntax



Parameters

-access

Specifies the access flag which gives the user write access to a consistent secondary. This enables write access to the secondary VDisks in the group if the group is in a consistent state.

rc_consist_group_id | rc_consist_group_name

Specifies the ID or name of the consistency group to stop.

Description

This command applies to a consistency group. You can issue this command to stop a consistency group that is copying from primary to secondary.

If the consistency group is in an inconsistent state, any copy operation stops and will not resume until you issue the **svctask startrconsistgrp** command. Write activity will no longer be copied from the primary to the secondary virtual disks belonging to the relationships in the group. For a consistency group in the ConsistentSynchronized state, this command causes a consistency freeze.

When a consistency group is in a consistent state (for example, in the ConsistentStopped, ConsistentSynchronized, or ConsistentDisconnected state) then the **-access** argument may be used with the **stoprconsistgrp** command to enable write access to the secondary virtual disks within that group.

Initial state	Final state	Notes
InconsistentStopped	InconsistentStopped	-
InconsistentCopying	InconsistentStopped	-
ConsistentStopped	ConsistentStopped	-access permitted
ConsistentSynchronized	ConsistentStopped	-access permitted
Idling	ConsistentStopped	-access permitted
IdlingDisconnected	unchanged	A relationship may move to stopped state when reconnected.
InconsistentDisconnected	InconsistentStopped	On the cluster issuing the svctask stoprconsistgrp command.
InconsistentDisconnected	unchanged	On the disconnected cluster.

Initial state	Final state	Notes
ConsistentDisconnected	ConsistentStopped	On the cluster issuing the svctask stopprconsistgrp command, -access permitted.
ConsistentDisconnected	unchanged	On the disconnected cluster, -access permitted.

Possible failures

- CMMVC5786E The action failed because the cluster is not in a stable state.
- CMMVC5936E The action failed because an entity that was specified in the command does not exist.

An invocation example

```
svctask stopprconsistgrp rccopy1
```

The resulting output

No feedback

stopprrelationship

You can use the **stopprrelationship** command to stop the copy process for a Mirror relationship. It may also be used to enable write access to a consistent secondary VDisk.

Syntax

```
svctask -- stopprrelationship -- [-access] [rc_rel_id | rc_rel_name]
```

Parameters

-access

Specifies the allow access flag which gives the user write access to a consistent secondary.

rc_rel_id | rc_rel_name

Specifies the ID or name of the relationship to stop. Specify only for stand-alone relationships.

Description

This command applies to a standalone relationship. It will be 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.

If the relationship is in an inconsistent state, any copy operation stops and will not resume until you issue a **svctask startprrelationship** command. Write activity will no longer be copied from the primary to the secondary virtual disk. For a relationship in the ConsistentSynchronized state, this command causes a consistency freeze.

When a relationship is in a consistent state (ie, in the ConsistentStopped, ConsistentSynchronized, or ConsistentDisconnected state) then the -access

argument may be used with the stopprrelationship command to enable write access to the secondary virtual disk.

Initial state	Final state	Notes
InconsistentStopped	InconsistentStopped	-
InconsistentCopying	InconsistentStopped	-
ConsistentStopped	ConsistentStopped	-access permitted
ConsistentSynchronized	ConsistentStopped	-access permitted
Idling	ConsistentStopped	-access permitted
IdlingDisconnected	unchanged	A relationship may move to stopped state when reconnected.
InconsistentDisconnected	InconsistentStopped	On the cluster issuing the svctask stopprrelationship command.
InconsistentDisconnected	unchanged	On the disconnected cluster.
ConsistentDisconnected	ConsistentStopped	On the cluster issuing the svctask stopprrelationship command, -access permitted.
ConsistentDisconnected	unchanged	On the disconnected cluster, -access permitted.

Possible failures

- CMMVC5786E The action failed because the cluster is not in a stable state.
- CMMVC5936E The action failed because an entity that was specified in the command does not exist.

An invocation example

```
svctask stopprrelationship rccopy1
```

The resulting output

No feedback

switchrconsistgrp

You can use the **switchrconsistgrp** command to reverse the roles of primary and secondary virtual disks in a Mirror consistency group when that consistency group is in a consistent state. This change will be applied to all the relationships in the consistency group.

Syntax

```

▶▶▶ svctask — — switchrconsistgrp — — -primary ———— master —————▶
                                     └─ aux ─────────┘

▶└─ rc_consist_group_id ─────────────────────────────────────────────────▶
  └─ rc_consist_group_name ───────────────────────────────────────────▶

```

Parameters

-primary *master* | *aux*

Specifies whether the master or auxiliary side of the relationships in the group will become the primary VDisks.

rc_consist_group_id | **rc_consist_group_name**

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 in a consistency group, perhaps as part of a graceful failover. Write access to the former primary VDisks is lost and write access to the new primary VDisks is acquired. This command will only be successful when the consistency group is in a connected, consistent state, and when reversing the direction of the relationships would not lead to a loss of consistency, for example, when the consistency group is consistent and synchronized. This command will therefore only succeed when the consistency group is in one of the following states:

- ConsistentSynchronized
- ConsistentStopped and Synchronized
- Idling and Synchronized

The consistency group moves to the ConsistentSynchronized state after successful completion of this command. If you specify the **-primary** argument with the current primary, the command has no affect.

Possible failures

- CMMVC5786E The action failed because the cluster is not in a stable state.
- CMMVC5936E The action failed because an entity that was specified in the command does not exist.

An invocation example

```
svctask switchrcconsistgrp -primary aux rccopy2
```

The resulting output

No feedback

switchcrelationship

Use the **switchcrelationship** to reverse the roles of primary and secondary virtual disks in a Mirror relationship when that relationship is in a consistent state.

Syntax

```
svctask switchcrelationship -primary master | aux  
rc_rel_id | rc_rel_name
```

Parameters

-primary *master* | *aux*

Specifies whether the master or auxiliary is to be the primary.

rc_rel_id | rc_rel_name

Specifies the ID or name of the relationship to switch.

Description

This command applies to a stand-alone relationship. It will be 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 virtual disk in a relationship or consistency group, perhaps as part of a graceful failover. Write access to the old primary is lost and write access to the new primary is acquired. This command will only be successful when the relationship is in a connected, consistent state, and when reversing the direction of the relationship would not lead to a loss of consistency; that is, when the relationship is consistent and synchronized. This command will, therefore, only succeed when the relationship is in one of the following states:

- ConsistentSynchronized
- ConsistentStopped and Synchronized
- Idling and Synchronized

The relationship moves to the ConsistentSynchronized state after successful completion of this command. If you specify the `-primary` argument with the current primary, the command has no effect.

Possible failures

- CMMVC5786E The action failed because the cluster is not in a stable state.
- CMMVC5936E The action failed because an entity that was specified in the command does not exist.

An invocation example

```
svctask switchrcrelationship -primary master rccopy2
```

The resulting output

No feedback

Chapter 4. Information commands

The following commands enable you to work with displaying specific types of information with the SAN Volume Controller.

Note: IDs are assigned at run-time by the system and cannot be relied upon to be the same after configuration restoration. Therefore, wherever possible, object names should be used in preference to IDs when working with objects.

lscluster

The **lscluster** command returns a report as a concise list, or a detailed view, of clusters. The list provides possible values that are applicable to the attributes that are displayed as data in the output views.

The list report style can be used to obtain two styles of report.

1. A list containing concise information about all clusters. (Each entry in the list corresponds to a single cluster.)
2. The detailed information about a single, user specified, cluster.

Syntax

```
➤— svcinfo — — lscluster — [ -filtervalue — attrib=value ] —————➤
➤ [ -nohdr ] [ -bytes ] [ -delim — delimiter ] —————➤
➤ [ object_id ] [ -filtervalue? ] —————➤
   [ name ]
```

Parameters

-filtervalue *attribute=value*

Optionally 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.

-nohdr

By default, headings are displayed for each column of data (in a concise style view), and each item of data (in a detailed style view). Using the **-nohdr** parameter will suppress the display of these headings.

Note: If there is no data to be displayed (for example, an empty view has been returned), then headings are not displayed irrespective of whether the **-nohdr** option was used or not.

-bytes

Optionally used to display all capacities as bytes.

-delim *delimiter*

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 will override this behavior. Valid input for the `-delim` parameter is a one byte character. If, for example, you entered `-delim :` a colon will be used to separate all items of data in a concise view (for example, the spacing of columns does not occur) and in a detailed view the data is separated from its header by a colon.

object_id | name

Optionally specifies the name or ID of an object. If not supplied, the concise view of all objects, or all of those objects matching the filtering requirements in `-filtervalue` (if specified), of the given type are returned. If supplied, the detailed view of the specific object is returned, and any `filtervalue` entry (if entered) is ignored.

-filtervalue?

Display a list of valid filter attributes. The valid filters for the `svcinfo lscluster` command are:

- cluster_name
- cluster_unique_id
- id
- name

Description

This command will return a concise list, or a detailed view, of clusters.

The following list provides possible values that are applicable to the attributes that are displayed as data in the output views:

location	local, remote
statistics status	on, off
SNMP setting	none, all, hardware_only

The location, partnership and bandwidth fields are relevant to Metro or Global Mirror configurations where the SAN fabrics of two clusters are linked together. Information about the remote cluster will be reported by the `lscluster` command if the `mkpartnership` command has been issued from the local cluster to the remote cluster. For example, if the partnership has been at least partially established from the local cluster.

You can issue the `svcinfo lscluster` command to display a concise view of the cluster.

```
svcinfo lscluster -delim : 10030a007e5
```

where `10030a007e5` is the name of the cluster. The output from this command will include the following for each cluster on the fabric:

- cluster name
- cluster IP address
- cluster service mode IP address

For the remote cluster, these fields indicate the following:

- location: remote

- partnership:
 - partially_configured (mkpartnership command has only been issued from the local cluster to the remote cluster)
 - fully_configured (mkpartnership command has been issued in both directions)
- bandwidth: MB/sec (the bandwidth available on the inter-cluster link for background copy)

Possible failures

- CMMVC5786E The action failed because the cluster is not in a stable state.
- CMMVC5804E The action failed because an entity that was specified in the command does not exist.

A concise invocation example

```
svcinfo lscluster -delim :
```

The concise resulting output

```
id:name:location:partnership:bandwidth:cluster_IP_address:
id:name:location:partnership:bandwidth:cluster_IP_address:
cluster_service_ip_address:id_alias
0000020062813ABA:clusterA:local:::9.20.247.210:1.1.1.1:0000020062813ABA
0000020062006746:clusterB:remote:fully_configured:50:9.20.247.211:
1.1.1.1:0000020062006746
```

A detailed invocation example

```
svcinfo lscluster -delim : 10030a007e5
```

The detailed resulting output

```
id:1521071282978998
name:cluster1
location:local
partnership:
bandwidth:
cluster_IP_address:9.20.165.16
cluster_service_IP_address:9.20.165.17total_mdisk_capacity:59.8GB
space_in_mdisk_grps:0
space_allocated to vdisks:0
total_free_space:59.8GB
statistics_status:on
statistics_frequency:300
required_memory:1280
subnet_mask:255.255.255.0
default_gateway:9.20.165.1
cluster_locale:en_US
SNMP_setting:snmp all
SNMP_community:
SNMP_server_IP_address:9.20.165.18
time_zone:522 UTCemail_setting:all
email_id:another@support.com
code_level:1.20abcG
FC_port_speed:1Gb
id_alias:1521071282978998
gm_link_tolerance:60
gm_inter_cluster_delay_simulation:0
gm_intra_cluster_delay_simulation:0
```

A concise invocation example for a Metro or Global Mirror configuration, where clusterA has issued the mkpartnership to clusterB, and the intercluster bandwidth is set to 50 MB/s.

```
svcinfo lscluster -delim :
```

The concise resulting output

```
id:name:location:partnership:bandwidth:
 cluster IP address:cluster service IP address
0000020062813ABA:clusterA:local:::9.20.247.210:1.1.1.1
0000020062006746:clusterB:remote:
 fully_configured:50:9.20.247.211:1.1.1.1
```

lsclustercandidate

The **lsclustercandidate** command lists the clusters that are available for setting up a two-cluster partnership. This is a prerequisite for creating inter-cluster Metro or Global Mirror relationships.

Syntax

```
▶▶ svcinfo — lsclustercandidate — [ -nohdr ] —————▶
▶ [ -delim — delimiter ] —————▶▶
```

Parameters

-nohdr

By default, headings are displayed for each column of data (in a concise style view), and each item of data (in a detailed style view). Using the `-nohdr` parameter will suppress the display of these headings.

Note: If there is no data to be displayed (for example, an empty view has been returned), then headings are not displayed irrespective of whether the `-nohdr` option was used or not.

-delim *delimiter*

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 will override this behavior. Valid input for the `-delim` parameter is a one byte character. If, for example, you entered `-delim :` a colon will be used to separate all items of data in a concise view (for example, the spacing of columns does not occur) and in a detailed view the data is separated from its header by a colon.

Description

This command returns a list of clusters that are available as candidate partner clusters to form a Metro or Global Mirror Partnership between two clusters.

Possible failures

- CMMVC5786E The action failed because the cluster is not in a stable state.
- CMMVC5804E The action failed because an entity that was specified in the command does not exist.

An invocation example

```
svcinfo lsclustercandidate
```

The resulting output

```
id            configured  cluster_name
0000010034E0F430 no                1dcluster26
```

Iserrlogbyrcrelationship

The **Iserrlogbyrcrelationship** command displays a list of the errors and events in the log that are related to Metro or Global Mirror relationships.

Syntax

```
svcinfo -- Iserrlogbyrcrelationship -- [ -count number ]
[ -config ] [ -unfixed ] [ -nohdr ] [ -delim delimiter ]
[ rrelationship_id | rrelationship_name ]
```

Parameters

-count *number*

Optionally specifies to only list the last number entries in the log. The **-count** argument specifies the maximum number of errors or events to list.

-config

Optionally specifies to only list configuration events. When **-config** is passed, the command will act as described above, but only list configuration events.

-unfixed

Optionally specifies to only list unfixed errors. When **-unfixed** is passed, the command will act as above, but will only list unfixed errors.

-nohdr

By default, headings are displayed for each column of data (in a concise style view), and each item of data (in a detailed style view). Using the **-nohdr** parameter will suppress the display of these headings.

Note: If there is no data to be displayed (for example, an empty view has been returned), then headings are not displayed irrespective of whether the **-nohdr** option was used or not.

-delim *delimiter*

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 will override this behavior. Valid input for the **-delim** parameter is a one byte character. If, for example, you entered **-delim :** a colon will be used to separate all items of data in a concise view (for example, the spacing of columns does not occur) and in a detailed view the data is separated from its header by a colon.

rrelationship_id | **rrelationship_name**

Optionally specifies the object id used to filter the log.

Description

When issued, this command will display a list of the errors and events in the log related to Metro or Global Mirror relationships. The list can be filtered further by specifying a specific object ID or name. This will return only the errors and events that have been logged against the specified object. The list can also be filtered to show only the configuration events or the unfixed errors for the given object type or object ID. Similarly the last x entries against a given object type or object ID can be listed.

Note: Although there is an object type of unknown displayed in the error log, there is no command with which to filter these object types.

Possible failures

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

An invocation example

```
svcinfo lserrlogbyrcrelationship -delim :
```

The resulting output

```
id:type:fixed:SNMP_trap_raised:error_type:node_name:sequence_number:
root_sequence_number:first_timestamp:last_timestamp:number_of_errors:error_code
2:remote:no:no:5:node1:0:0:030407090442:030407090442:1:00990226
2:remote:no:no:5:node1:0:0:030407090106:030407090106:1:00990225
1:remote:no:no:5:node1:0:0:030407085932:030407085932:1:00990225
2:remote:no:no:6:n/a:106:106:030407090117:030407090117:1:00985002
```

lslicense

The **lslicense** command returns the current license (featurization) settings for the cluster. The settings are defined as Copy Services status and the capacity of virtual storage licensed for use by this cluster.

Syntax

```
➤➤ svcinfo -- lslicense -- [-nohdr] [-delim delimiter]
```

Parameters

-nohdr

By default, headings are displayed for each column of data (in a concise style view), and each item of data (in a detailed style view). Using the **-nohdr** parameter will suppress the display of these headings.

Note: If there is no data to be displayed (for example, an empty view has been returned), then headings are not displayed irrespective of whether the **-nohdr** option was used or not.

-delim *delimiter*

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 will override this behavior. Valid input for the **-delim** parameter is a one byte character. If, for example, you entered **-delim :** a colon

will be used to separate all items of data in a concise view (for example, the spacing of columns does not occur) and in a detailed view the data is separated from its header by a colon.

Description

This command returns the licensed features of the cluster. That is, the copy services status and the capacity of virtual storage licensed for use by this cluster.

You can issue the **svcinfolicense** command to return the current license (featurization) settings for the cluster. You can issue the **svctask chlicense** command to change the licensed settings of the cluster. Because the feature settings are entered when the cluster is first created, you need to update the settings only if you have changed your license. You can change the following values:

- FlashCopy: disabled or enabled
- Remote copy (Global or Metro Mirror): disabled or enabled
- Virtualization limit: number, in gigabytes (1073741824 bytes)

The output displayed lists the feature functions in a list and displays whether they are enabled or disabled.

Possible failures

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

An invocation example

```
svcinfolicense
```

The resulting output

```
feature_flash on
feature_remote on
feature_num_gb 32
```

lsrconsistgrp

The **lsrconsistgrp** command will return a concise list, or a detailed view, of Metro or Global Mirror consistency groups visible to the cluster.

The list report style can be used to obtain two styles of report.

1. The attributes of the Metro or Global Mirror consistency group, plus the ID and name of every relationship that is in the group.
2. The detailed information about a single Metro or Global Mirror consistency group.

Syntax

```
svcinfolicense -lsrconsistgrp [-filtervalue -attrib=value]
                             [-nohdr] [-delim delimiter] [-object_id name]
```

└─ filtervalue? ─┘

Parameters

-filtervalue *attribute=value*

Optionally 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.

-nohdr

By default, headings are displayed for each column of data (in a concise style view), and each item of data (in a detailed style view). Using the **-nohdr** parameter will suppress the display of these headings.

Note: If there is no data to be displayed (for example, an empty view has been returned), then headings are not displayed irrespective of whether the **-nohdr** option was used or not.

-delim *delimiter*

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 will override this behavior. Valid input for the **-delim** parameter is a one byte character. If, for example, you entered **-delim :** a colon will be used to separate all items of data in a concise view (for example, the spacing of columns does not occur) and in a detailed view the data is separated from its header by a colon.

object_id | name

Optionally specifies the name or ID of an object. If not supplied, the concise view of all objects, or all of those objects matching the filtering requirements in **-filtervalue** (if specified), of the given type are returned. If supplied, the detailed view of the specific object is returned, and any **filtervalue** entry (if entered) is ignored.

-filtervalue?

Display a list of valid filter attributes. The valid filters for the **svcinfo lsrrconsistgrp** command are:

- 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 will return a concise list, or a detailed view, of Metro or Global Mirror consistency groups that are visible to the cluster.

The following list provides possible values that are applicable to the attributes that are displayed as data in the output views:

primary	n/a, master, aux
state	inconsistent_stopped, inconsistent_copying, consistent_stopped, consistent_synchronized, idling, idling_disconnected, inconsistent_disconnected, consistent_disconnected, empty
freeze_time	The time in YY/MM/DD/HH/MM format.
status	online, primary_offline, secondary_offline
sync	in_sync, out_of_sync
copy_type	empty_group, metro, global

Note: The names of the Metro or Global Mirror relationships and consistency groups might be blank if the relationship or consistency groups are inter-cluster and the cluster partnership is disconnected.

Possible failures

- CMMVC5786E The action failed because the cluster is not in a stable state.
- CMMVC5804E The action failed because an entity that was specified in the command does not exist.

A concise invocation example

```
svcinfo lsrrconsistgrp -delim :
```

The concise resulting output

```
id:name:master_cluster_id:master_cluster_name:aux_cluster_id:aux_cluster_name:primary:state:relationship_count
248:jdemo_BA_cons1:0000020060406746:clusterB:0000020061413ABA:clusterA:master:consistent_stopped:2
249:rccstgrp0:0000020061413ABA:clusterA:0000020061413ABA:clusterA::empty:0
250:jdemo_BA_cons2:0000020060406746:clusterB:0000020061413ABA:clusterA:master:inconsistent_stopped:1
251:BA_cons1:0000020060406746:clusterB:0000020061413ABA:clusterA:master:consistent_stopped:4
252:AB_cons2:0000020061413ABA:clusterA:0000020060406746:clusterB::empty:0
253:AB_cons1:0000020061413ABA:clusterA:0000020060406746:clusterB:aux:consistent_stopped:3
254:AA_cons2:0000020061413ABA:clusterA:0000020061413ABA:clusterA::empty:0
255:AA_cons1:0000020061413ABA:clusterA:0000020061413ABA:clusterA:master:consistent_synchronized:2
```

A detailed invocation example

```
svcinfo lsrrconsistgrp -delim : 254
```

The detailed resulting output

```
id:254
name:rccstgrp0
master_cluster_id:0000010030A007E5
master_cluster_name:kkk
aux_cluster_id:0000010030A007E5
aux_cluster_name:kkk
```

```

primary:master
state:inconsistent_stopped
relationship_count:1
freeze_time:
status:online
sync:
copy_type:metro
RC_rel_id:2
RC_rel_name:aaa

```

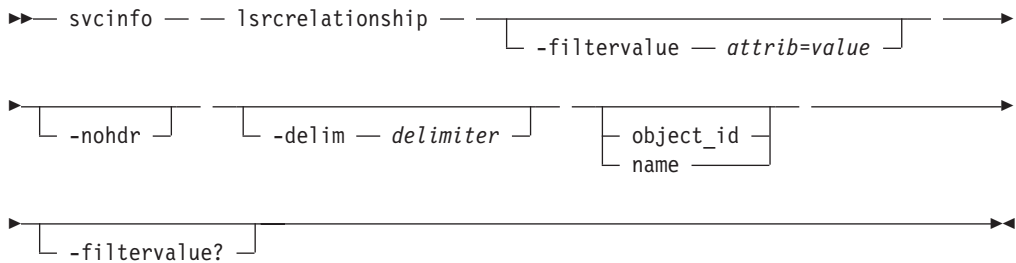
lsrcrelationship

The **lsrcrelationship** command will return a concise list, or a detailed view, of Metro or Global Mirror relationships visible to the cluster.

The list report style can be used to obtain two styles of report.

1. A list containing concise information about all the Metro or Global Mirror relationships visible to the cluster. (Each entry in the list corresponds to a single Metro or Global Mirror relationship.)
2. The detailed information about a single Metro or Global Mirror relationship.

Syntax



Parameters

-filtervalue *attribute=value*

Optionally 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.

-nohdr

By default, headings are displayed for each column of data (in a concise style view), and each item of data (in a detailed style view). Using the **-nohdr** parameter will suppress the display of these headings.

Note: If there is no data to be displayed (for example, an empty view has been returned), then headings are not displayed irrespective of whether the **-nohdr** option was used or not.

-delim *delimiter*

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 will override this behavior. Valid input for the **-delim** parameter is a one byte character. If, for example, you entered **-delim :** a colon

will be used to separate all items of data in a concise view (for example, the spacing of columns does not occur) and in a detailed view the data is separated from its header by a colon.

object_id | name

Optionally specifies the name or ID of an object. If not supplied, the concise view of all objects, or all of those objects matching the filtering requirements in -filtervalue (if specified), of the given type are returned. If supplied, the detailed view of the specific object is returned, and any filtervalue entry (if entered) is ignored.

-filtervalue?

Display a list of valid filter attributes. The valid filters for the **svcinfolsrcrelationship** command are:

- RC_rel_id
- RC_rel_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
- progress

Description

This command will return a concise list or a detailed view, of Metro or Global Mirror relationships visible to the cluster.

The following list provides possible values that are applicable to the attributes that are displayed as data in the output views:

primary	n/a, master, aux
state	inconsistent_stopped, inconsistent_copying, consistent_stopped, consistent_synchronized, idling, idling_disconnected, inconsistent_disconnected, consistent_disconnected
progress	0-100, n/a
freeze time	The time in YY/MM/DD/HH/MM format.
status	online, primary_offline, secondary_offline
sync	n/a, in_sync, out_of_sync
copy_type	Specifies the mirror copy type, either metro or global.

Note: The names of the Metro or Global Mirror relationships and consistency groups may be blank if the relationship or consistency groups are inter-cluster and the cluster partnership is disconnected.

Possible failures

- CMMVC5786E The action failed because the cluster is not in a stable state.
- CMMVC5804E The action failed because an entity that was specified in the command does not exist.

A concise and detailed invocation example

```
svcinfo lsrelationship -delim : -filtervalue name=j*
```

The concise and detailed 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
| 45:jrel_AB1:0000020061413ABA:clusterA:45:jdisk_B8:0000020060406746:clusterB:38:j
| disk_B1:master:::consistent_stopped:50:metro
| 48:jrel_AB2:0000020061413ABA:clusterA:48:jdisk_A4:0000020060406746:clusterB:41:j
| disk_B4:master:::consistent_synchronised:50:metro
| 49:jrel_BA_1:0000020060406746:clusterB:42:jdisk_B5:0000020061413ABA:clusterA:49:j
| disk_A5:master:248:jdemo_BA_cons1:consistent_stopped:50:metro
| 50:jrel_BA_2:0000020060406746:clusterB:43:jdisk_B6:0000020061413ABA:clusterA:
| 50:jdisk_A6:master:248:jdemo_BA_cons1:consistent_stopped:50:metro
```

A detailed invocation example

```
svcinfo lsrelationship -delim : AB_2
```

The detailed resulting output

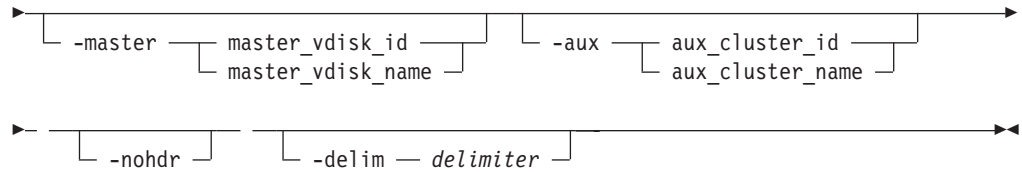
```
| id:9
| name:AB_2
| master_cluster_id:0000020061413ABA
| master_cluster_name:clusterA
| master_vdisk_id:9
| master_vdisk_name:stripe9
| aux_cluster_id:0000020060406746
| aux_cluster_name:clusterB
| aux_vdisk_id:9
| aux_vdisk_name:stripe9_b
| primary:master
| consistency_group_id:
| consistency_group_name:
| state:consistent_stopped
| bg_copy_priority:50
| progress:
| freeze_time:2006/05/05/08/26/46
| status:secondary_offline
| sync:in_sync
| copy_type:metro
```

lsrelationshipcandidate

The **lsrelationshipcandidate** command lists VDisks that are eligible to form Metro or Global Mirror relationships. You can list suitable VDisks on the local or remote cluster.

Syntax

```
▶— svcinfo — — lsrelationshipcandidate —————▶
```



Parameters

-master *master_vdisk_id* | *master_vdisk_name*

You can use this parameter to specify a particular vdisk that you want to use as the master vdisk. The command will then look for candidates that match the size of this vdisk. If you are requesting candidate vdisks on the local cluster, this command will also match the `io_group`.

-aux *aux_cluster_id* | *aux_cluster_name*

Specifies a remote cluster to find vdisk candidates for an inter-cluster relationship. If you do not specify this parameter, the candidates on the local cluster are returned.

-nohdr

By default, headings are displayed for each column of data (in a concise style view), and each item of data (in a detailed style view). Using the `-nohdr` parameter will suppress the display of these headings.

Note: If there is no data to be displayed (for example, an empty view has been returned), then headings are not displayed irrespective of whether the `-nohdr` option was used or not.

-delim *delimiter*

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 will override this behavior. Valid input for the `-delim` parameter is a one byte character. If, for example, you entered `-delim :` a colon will be used to separate all items of data in a concise view (for example, the spacing of columns does not occur) and in a detailed view the data is separated from its header by a colon.

Description

This command returns a list of VDisks that can be the master or auxiliary disk for a Metro or Global Mirror relationship. VDisk IDs and names are returned.

Possible failures

- CMMVC5786E The action failed because the cluster is not in a stable state.
- CMMVC5804E The action failed because an entity that was specified in the command does not exist.

An invocation example

```
svcinfo lsrelationshipcandidate -delim :
```

The resulting output

```
id:vdisk_name
0:vdisk0
4:vdisk4
```

lsrrelationshipprogress

You can use the **lsrrelationshipprogress** command to return the progress of the background copy of a Metro or Global Mirror relationship. When the initial background copy process for a relationship has completed, null will be displayed for the progress of that relationship.

Syntax

```
svcinfo -- lsrrelationshipprogress -- [ -nohdr ] [ -delim delimiter ] [ rrelationship_id ] [ rrelationship_name ]
```

Parameters

-nohdr

By default, headings are displayed for each column of data (in a concise style view), and each item of data (in a detailed style view). Using the `-nohdr` parameter will suppress the display of these headings.

Note: If there is no data to be displayed (for example, an empty view has been returned), then headings are not displayed irrespective of whether the `-nohdr` option was used or not.

-delim *delimiter*

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 will override this behavior. Valid input for the `-delim` parameter is a one byte character. If, for example, you entered `-delim :` a colon will be used to separate all items of data in a concise view (for example, the spacing of columns does not occur) and in a detailed view the data is separated from its header by a colon.

rrelationship_id | *rrelationship_name*

Specifies the specific object ID or name of the given type.

Description

This command returns the percentage progress of the background copy of a Metro or Global Mirror relationship.

Possible failures

- CMMVC5786E The action failed because the cluster is not in a stable state.
- CMMVC5804E The action failed because an entity that was specified in the command does not exist.

An invocation example

```
svcinfo lsrrelationshipprogress -delim : 0
```

The resulting output

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

Notices

This information was developed for products and services offered in the U.S.A.

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 give 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.*

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATIONS "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 states 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 Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been

estimated through extrapolation. Actual results may vary. Users of this document may verify the applicable data for their specific environment.

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 may be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

- Enterprise Storage Server
- FlashCopy
- IBM
- IBM TotalStorage
- IBM System Storage
- System Storage
- TotalStorage

Intel and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft and Windows 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.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

Glossary

This is the glossary for the SAN Volume Controller.

A

application server

A host that is attached to the storage area network (SAN) and that runs applications.

asymmetric virtualization

A virtualization technique in which the virtualization engine is outside the data path and performs a metadata-style service. The metadata server contains all the mapping and locking tables while the storage devices contain only data. See also *symmetric virtualization*.

auxiliary virtual disk

The virtual disk that contains a backup copy of the data and that is used in disaster recovery scenarios. See also *master virtual disk*.

C

cluster

In SAN Volume Controller, a pair of nodes that provides a single configuration and service interface.

configuration node

A node that acts as the focal point for configuration commands and manages the data that describes the cluster configuration.

consistency group

A group of copy relationships between virtual disks that are managed as a single entity.

consistent copy

In a Metro or Global Mirror relationship, a copy of a secondary virtual disk (VDisk) that is identical to the primary VDisk from the viewpoint of a host system, even if a power failure occurred while I/O activity was in progress.

copied

In a FlashCopy relationship, a state that indicates that a copy has been started after the copy relationship was created. The copy process is complete and the target disk has no further dependence on the source disk.

copying

A status condition that describes the state of a pair of virtual disks (VDisks) that have a copy relationship. The copy process has been started but the two virtual disks are not yet synchronized.

counterpart SAN

A nonredundant portion of a redundant storage area network (SAN). A counterpart SAN provides all the connectivity of the redundant SAN but without the redundancy. Each counterpart SANs provides an alternate path for each SAN-attached device. See also *redundant SAN*.

D

data migration

The movement of data from one physical location to another without disrupting I/O operations.

degraded

Pertaining to a valid configuration that has suffered a failure but continues to be supported and legal. Typically, a repair action can be performed on a degraded configuration to restore it to a valid configuration.

dependent write operations

A set of write operations that must be applied in the correct order to maintain cross-volume consistency.

destage

A write command initiated by the cache to flush data to disk storage.

directed maintenance procedures

The set of maintenance procedures that can be run for a cluster. These procedures are run from within the SAN Volume Controller application and are documented in the service guide.

disconnected

In a Metro or Global Mirror relationship, pertains to two clusters when they cannot communicate.

disk controller

A device that coordinates and controls the operation of one or more disk drives and synchronizes the operation of the drives with the operation of the system as a whole. Disk controllers provide the storage that the cluster detects as managed disks (MDisks).

disk zone

A zone defined in the storage area network (SAN) fabric in which the SAN Volume Controller can detect and address the logical units that the disk controllers present.

E

error code

A value that identifies an error condition.

ESS See *IBM TotalStorage[®] Enterprise Storage Server[®]*.

exclude

To remove a managed disk (MDisk) from a cluster because of certain error conditions.

excluded

In SAN Volume Controller, the status of a managed disk that the cluster has removed from use after repeated access errors.

extent A unit of data that manages the mapping of data between managed disks and virtual disks.

F

failover

In SAN Volume Controller, the function that occurs when one redundant part of the system takes over the workload of another part of the system that has failed.

fibre channel

A technology for transmitting data between computer devices at a data rate of up to 4 Gbps. It is especially suited for attaching computer servers to shared storage devices and for interconnecting storage controllers and drives.

FC See *fibre channel*.

FlashCopy service

In SAN Volume Controller, a copy service that duplicates the contents of a source virtual disk (VDisk) to a target VDisk. In the process, the original contents of the target VDisk are lost. See also *point-in-time copy*.

FlashCopy mapping

A relationship between two virtual disks.

FlashCopy relationship

See *FlashCopy mapping*.

G**Global Mirror**

An asynchronous copy service that enables host data on a particular source virtual disk (VDisk) to be copied to the target VDisk that is designated in the relationship.

H

HBA See *host bus adapter*.

host bus adapter (HBA)

In SAN Volume Controller, an interface card that connects a host bus, such as a peripheral component interconnect (PCI) bus, to the storage area network.

host An open-systems computer that is connected to the SAN Volume Controller through a fibre-channel interface.

host ID

In SAN Volume Controller, a numeric identifier assigned to a group of host fibre-channel ports for the purpose of logical unit number (LUN) mapping. For each host ID, there is a separate mapping of Small Computer System Interface (SCSI) IDs to virtual disks (VDisks).

host zone

A zone defined in the storage area network (SAN) fabric in which the hosts can address the SAN Volume Controllers.

I**IBM TotalStorage Enterprise Storage Server (ESS)**

An IBM product that provides an intelligent disk-storage subsystem across an enterprise.

idling

- The status of a pair of virtual disks (VDisks) that have a defined copy relationship for which no copy activity has yet been started.
- In a Metro or Global Mirror relationship, the state that indicates that the master virtual disks (VDisks) and auxiliary VDisks are operating in the primary role. Consequently, both VDisks are accessible for write I/O operations.

illegal configuration

A configuration that will not operate and will generate an error code to indicate the cause of the problem.

image mode

An access mode that establishes a one-to-one mapping of extents in the managed disk (MDisk) with the extents in the virtual disk (VDisk). See also *managed space mode* and *unconfigured mode*.

image VDisk

A virtual disk (VDisk) in which there is a direct block-for-block translation from the managed disk (MDisk) to the VDisk.

inconsistent

In a Metro or Global Mirror relationship, pertaining to a secondary virtual disk (VDisk) that is being synchronized with the primary VDisk.

input/output (I/O)

Pertaining to a functional unit or communication path involved in an input process, an output process, or both, concurrently or not, and to the data involved in such a process.

integrity

The ability of a system to either return only correct data or respond that it cannot return correct data.

Internet Protocol (IP)

In the Internet suite of protocols, a connectionless protocol that routes data through a network or interconnected networks and acts as an intermediary between the higher protocol layers and the physical network.

I/O See *input/output*.

I/O group

A collection of virtual disks (VDisks) and node relationships that present a common interface to host systems.

I/O throttling rate

The maximum rate at which an I/O transaction is accepted for this virtual disk (VDisk).

IP See *Internet Protocol*.

L

LBA See *logical block address*.

local fabric

In SAN Volume Controller, those storage area network (SAN) components (such as switches and cables) that connect the components (nodes, hosts, switches) of the local cluster together.

local/remote fabric interconnect

The storage area network (SAN) components that are used to connect the local and remote fabrics together.

logical block address (LBA)

The block number on a disk.

logical unit (LU)

An entity to which Small Computer System Interface (SCSI) commands are addressed, such as a virtual disk (VDisk) or managed disk (MDisk).

logical unit number (LUN)

The SCSI identifier of a logical unit within a target. (S)

LU See *logical unit*.

LUN See *logical unit number*.

M**managed disk (MDisk)**

A Small Computer System Interface (SCSI) logical unit that a redundant array of independent disks (RAID) controller provides and a cluster manages. The MDisk is not visible to host systems on the storage area network (SAN).

managed disk group

A collection of managed disks (MDisks) that, as a unit, contain all the data for a specified set of virtual disks (VDisks).

managed space mode

An access mode that enables virtualization functions to be performed. See also *image mode* and *unconfigured mode*.

mapping

See *FlashCopy mapping*.

master virtual disk

The virtual disk (VDisk) that contains a production copy of the data and that an application accesses. See also *auxiliary virtual disk*.

MDisk

See *managed disk*.

Metro Mirror

A synchronous copy service that enables host data on a particular source virtual disk (VDisk) to be copied to the target VDisk that is designated in the relationship.

migration

See *data migration*.

mirrorset

- IBM definition: See *RAID-1*.
- HP definition: A RAID storageset of two or more physical disks that maintain a complete and independent copy of the data from the virtual disk. This type of storageset has the advantage of being highly reliable and extremely tolerant of device failure. Raid level 1 storagesets are referred to as mirrorsets.

N

node One SAN Volume Controller. Each node provides virtualization, cache, and Copy Services to the storage area network (SAN).

node rescue

In SAN Volume Controller, the process by which a node that has no valid software installed on its hard disk drive can copy the software from another node connected to the same fibre-channel fabric.

O

offline

Pertaining to the operation of a functional unit or device that is not under the continual control of the system or of a host.

online Pertaining to the operation of a functional unit or device that is under the continual control of the system or of a host.

P

partnership

In Metro or Global Mirror, the relationship between two clusters. In a cluster partnership, one cluster is defined as the local cluster and the other cluster as the remote cluster.

paused

In SAN Volume Controller, the process by which the cache component quiesces all ongoing I/O activity below the cache layer.

pend To cause to wait for an event.

point-in-time copy

The instantaneous copy that the FlashCopy service makes of the source virtual disk (VDisk). In some contexts, this copy is known as a T_0 copy.

port The physical entity within a host, SAN Volume Controller, or disk controller system that performs the data communication (transmitting and receiving) over the fibre channel.

primary virtual disk

In a Metro or Global Mirror relationship, the target of write operations issued by the host application.

PuTTY

A client program that allows you to run remote sessions on your computer through specific network protocols, such as SSH, Telnet, and Rlogin.

Q

quorum disk

A managed disk (MDisk) that contains quorum data and that a cluster uses to break a tie and achieve a quorum.

R

RAID See *redundant array of independent disks*.

RAID 1

- SNIA dictionary definition: A form of storage array in which two or more identical copies of data are maintained on separate media.
- IBM definition: A form of storage array in which two or more identical copies of data are maintained on separate media. Also known as mirrorset.
- HP definition: See *mirrorset*.

RAID 5

- SNIA definition: A form of parity RAID in which the disks operate independently, the data strip size is no smaller than the exported block size, and parity check data is distributed across the array's disks. (S)
- IBM definition: See above.

- HP definition: A specially developed RAID storage set that stripes data and parity across three or more members in a disk array. A RAIDset combines the best characteristics of RAID level 3 and RAID level 5. A RAIDset is the best choice for most applications with small to medium I/O requests, unless the application is write intensive. A RAIDset is sometimes called parity RAID. RAID level 3/5 storage sets are referred to as RAIDsets.

RAID 10

A type of RAID that optimizes high performance while maintaining fault tolerance for up to two failed disk drives by striping volume data across several disk drives and mirroring the first set of disk drives on an identical set.

redundant array of independent disks

A collection of two or more disk drives that present the image of a single disk drive to the system. In the event of a single device failure, the data can be read or regenerated from the other disk drives in the array.

redundant SAN

A storage area network (SAN) configuration in which any one single component might fail, but connectivity between the devices within the SAN is maintained, possibly with degraded performance. This configuration is normally achieved by splitting the SAN into two, independent, counterpart SANs. See also *counterpart SAN*.

rejected

A status condition that describes a node that the cluster software has removed from the working set of nodes in the cluster.

relationship

In Metro or Global Mirror, the association between a master virtual disk (VDisk) and an auxiliary VDisk. These VDIs also have the attributes of a primary or secondary VDisk. See also *auxiliary virtual disk, master virtual disk, primary virtual disk, and secondary virtual disk*.

S

SAN See *storage area network*.

SAN Volume Controller fibre-channel port fan in

The number of hosts that can see any one SAN Volume Controller port.

SCSI See *Small Computer Systems Interface*.

secondary virtual disk

In Metro or Global Mirror, the virtual disk (VDisk) in a relationship that contains a copy of data written by the host application to the primary VDisk.

Secure Shell

A program to log in to another computer over a network, to execute commands in a remote machine, and to move files from one machine to another.

sequential VDisk

A virtual disk that uses extents from a single managed disk.

Simple Network Management Protocol (SNMP)

In the Internet suite of protocols, a network management protocol that is used to monitor routers and attached networks. SNMP is an

application-layer protocol. Information on devices managed is defined and stored in the application's Management Information Base (MIB).

Small Computer System Interface (SCSI)

A standard hardware interface that enables a variety of peripheral devices to communicate with one another.

SNMP

See *Simple Network Management Protocol*.

SSH See *Secure Shell*.

stand-alone relationship

In FlashCopy, Metro Mirror, and Global Mirror, relationships that do not belong to a consistency group and that have a null consistency group attribute.

stop A configuration command that is used to stop the activity for all copy relationships in a consistency group.

stopped

The status of a pair of virtual disks (VDisks) that have a copy relationship that the user has temporarily broken because of a problem.

storage area network (SAN)

A network whose primary purpose is the transfer of data between computer systems and storage elements and among storage elements. A SAN consists of a communication infrastructure, which provides physical connections, and a management layer, which organizes the connections, storage elements, and computer systems so that data transfer is secure and robust. (S)

subsystem device driver (SDD)

An IBM pseudo device driver designed to support the multipath configuration environments in IBM products.

superuser authority

The level of access required to add users.

suspended

The status of a pair of virtual disks (VDisks) that have a copy relationship that has been temporarily broken because of a problem.

symmetric virtualization

A virtualization technique in which the physical storage in the form of Redundant Array of Independent Disks (RAID) is split into smaller chunks of storage known as *extents*. These extents are then concatenated, using various policies, to make virtual disks (VDisks). See also *asymmetric virtualization*.

synchronized

In Metro or Global Mirror, the status condition that exists when both virtual disks (VDisks) of a pair that has a copy relationship contain the same data.

T

trigger

To initiate or reinitiate copying between a pair of virtual disks (VDisks) that have a copy relationship.

U

unconfigured mode

A mode in which I/O operations cannot be performed. See also *image mode* and *managed space mode*.

uninterruptible power supply

A device connected between a computer and its power source that protects the computer against blackouts, brownouts, and power surges. The uninterruptible power supply contains a power sensor to monitor the supply and a battery to provide power until an orderly shutdown of the system can be performed.

V

valid configuration

A configuration that is supported.

VDisk See *virtual disk*.

virtual disk (VDisk)

In SAN Volume Controller, a device that host systems attached to the storage area network (SAN) recognize as a Small Computer System Interface (SCSI) disk.

virtualization

In the storage industry, a concept in which a pool of storage is created that contains several disk subsystems. The subsystems can be from various vendors. The pool can be split into virtual disks that are visible to the host systems that use them.

virtualized storage

Physical storage that has virtualization techniques applied to it by a virtualization engine.

vital product data (VPD)

Information that uniquely defines system, hardware, software, and microcode elements of a processing system.

W

worldwide node name (WWNN)

An identifier for an object that is globally unique. WWNNs are used by Fibre Channel and other standards.

worldwide port name (WWPN)

A unique 64-bit identifier associated with a fibre-channel adapter port. The WWPN is assigned in an implementation- and protocol-independent manner.

WWNN

See *worldwide node name*.

WWPN

See *worldwide port name*.

Index

A

about this guide v
audience v

C

cluster
 commands 1
commands
 chcluster 1
 chpartnership 11
 chrconsistgrp 12
 chrrelationship 12
 expandvdisksize 5
 lscluster 31
 lsclustercandidate 34
 lserrlogbyrrelationship 35
 lslicense 36
 lsrconsistgrp 37
 lsrrelationship 40
 lsrrelationshipcandidate 42
 lsrrelationshipprogress 44
 mkrconsistgrp 15
 mkrrelationship 16
nodes
 Copy Services 11
 Global Mirror 11
 Metro Mirror 11
rmpartnership 19
rmrconsistgrp 20
rmrrelationship 21
rmvdisk 7
shrinkvdisksize 8
starttrconsistgrp 22
starttrrelationship 24
stopprconsistgrp 26
stopprrelationship 27
switchrconsistgrp 28
switchrrelationship 29
conventions v

D

data types ix
diagrams, syntax v

E

emphasis in text v
errata
 about this v

G

Global Mirror commands 11
glossary 47
guide
 about this v
 who should read v

I

information
 commands 31

M

Metro Mirror commands 11

N

notices
 legal 45

S

syntax diagrams v

T

terminology vii
text emphasis v
trademarks 46

V

value ranges ix
VDisks (virtual disks)
 commands 5
viewing
 clusters 31
 Global Mirror
 consistency groups 37
 relationships 40
 Metro Mirror
 consistency groups 37
 relationships 40

W

who should read this guide v
wildcards viii

