InfiniShell Commands Reference

Overview

The InfiniBox CLI comes with a rich, context-aware auto-complete, and with a simple and easy to use command structure that only takes administrators a few minutes to learn. With commands structured the same way across the board, and simple use of wildcards to apply the same commands to multiple objects, administrators love using InfiniShell to accelerate and simplify their work.

The InfinShell documentation is complemented by the InfiniBox user documentation:

- InfiniBox Management Console
- Accessing the InfiniShell Console
- InfiniShell session controls

General Commands

exit ALL ROLES

CHANGE

Description End current session.

Syntax

```
exit
```

Example

```
exit
```

Output

```
Goodbye!
Session closed.
```

help ALL ROLES

CHANGE

Description Displays help for commands, parameters and categories.

This command returns information on a searched command, category, parameter, string, or a list of commands than contain objects that match the search string.

Syntax

```
help [on=SUBJECT] [search=STR]
```

Arguments

on	Name of a command or a category
search	Any sequence of characters

In the following example we search for help on a snapshot.

Example

```
help search=snap
```

Output

```
Search results: snap

    Categories:

        cg.snap
        fs.snap
        vol.snap

    Commands:

        cg.snap.create                  source=CG name=NAME
                                        [snap_prefix=NAME]
                                        [snap_suffix=NAME]

        cg.snap.delete                  name=SG[,SG,...]
                                        [delete_members={YES|NO|Y|N}]

        cg.snap.member_query            name=SG[,SG,...]
                                        [--detailed]
                                        [--columns=FIELD[,FIELD,...]]
                                        [--sort=FIELD[,FIELD,...]]
                                        [--format={csv|json}]
                                        [--grep=PATTERN]
                                        [--grepv|--inverted-grep=PATTERN]
```

In the following example we search for help on the vol.create command.

Example

```
help on=vol.create
```

Output

```
Create a new volume.

    Syntax:

        vol.create name=NAME[,NAME,...] size=SIZE [pool=POOL]
                   [thin={YES|NO|Y|N}]
                   [ssd_cache={YES|NO|Y|N}]
                   [compression={YES|NO|Y|N}] [series=SERIES]

    Arguments:

        name                 A maximum of 65 Latin characters, numbers,
                             spaces, and the following symbols:
                             "^&'@()[]$=!-#{}%.+~_" (excluding quotation
                             marks). Leading and trailing whitespace
                             characters are stripped. (multiple values,
                             separated by commas).

        size                 A positive number (up to 2 digits beyond the
                             decimal point), optionally followed by a unit;
                             e.g. 1TB, 1000000000000.

        pool                 Name of an existing pool.

        thin                 Either yes, or no.

        ssd_cache            Either yes, or no.

        compression          Either yes, or no.

        series               Either a range of numbers, e.g "3-7", or a single
                             number, e.g "5", which is the same as "1-5".
                             Zero-padding the numbers yields a zero-padded
                             sequence. The series may contain at most 100
                             items, and the index of the last item may not
                             exceed 10000000000.
```

history ALL ROLES

CHANGE

Description Show user history.

Displays commands that were run by the user (by this user alone). Filterable by:

- `--grep`
- `--inverted-grep`

Syntax

```
history
```

Run the command with no arguments

Example

```
history
```

Output

```
--- COMMAND
1 pool.create name=p1 physical_capacity=10t
2 cg.create name=cg1
3 vol.create name=v-cg1- size=10g series=3
4 cg.add_member cg=cg1 member=v-cg-1,v-cg-2,v-cg-3
5 cg.member_query
6 history
```

Run the command looking for a specific string

Example

```
history --grep=cg
```

Output

```
--- COMMAND
2 cg.create name=cg1
3 vol.create name=v-cg1- size=10g series=3
4 cg.add_member cg=cg1 member=v-cg-1,v-cg-2,v-cg-3
5 cg.member_query
```

Run the command excluding a specific string

Example

```
history --inverted-grep=cg
```

Output

```
--- COMMAND
1 pool.create name=p1 physical_capacity=10t
6 history
```

login ALL ROLES

CHANGE

Description Login as another user.

Syntax

```
login [username=USERSTR]
```

Arguments

username	User name

Example

```
login username=admin
```

Output

```
The command prompt.
```

quit ALL ROLES

CHANGE

Description End current session.

Syntax

```
quit
```

Example

```
quit
```

Output

```
Goodbye!
Session closed.
```

Provisioning Commands

Chapter 1. Pool

Pool commands allow you to create a pool, to lock it for writes, to rename and resize it, to enable SSD cache, to set size thresholds for notifications and to query the InfiniBox on pools and their administrators.

pool.add_admin ADMIN

CHANGE

Description Grant pool provisioning privileges to a pool administrator.

Syntax

```
pool.add_admin pool=POOL user=POOLADMIN[,POOLADMIN,...]
```

Arguments

user	Name of an existing local user or ldap group, with PoolAdmin role (multiple values, separated by commas)
pool	Name of an existing pool

Example

```
pool.add_admin name=pool1 admin=pool-admin-may-17
```

Output

```
Granted provisioning privileges on pool pool1 to pool admins: local user pool-admin-may-17
```

pool.assign_qos ADMIN

CHANGE

Description Assign pool to policy.

Syntax

```
pool.assign_qos pool=POOL[,POOL,...] policy=QOSPOLICY
```

Arguments

pool	Name of an existing pool (multiple values, separated by commas)
policy	QoS policy

Example

```
pool.assign_qos pool=p1 policy=q3
```

Output

```
Pool 'p1' was assigned to policy 'q3'
```

pool.create ADMIN

CHANGE

Description Create a new pool.

Set the pool physical capacity, virtual capacity and emergency buffer. The emergency buffer determines whether the pool is allowed to extend without limitation, to extend within a predefined limitation, or not allowed to extend at all. The pool will extend into the emergency buffer once it passes its physical size. Then, the pool enters a Limited state where no new volumes can be created, but the existing volumes can be written into. If the pool can no longer expand, it enters a Locked state and its entities can no longer accept writes. When the pool is locked, no entity can be created and volumes cannot be resized. Release the Locked state (unlock the pool) using the pool.unlock command.

Syntax

```
pool.create name=NAME physical_capacity=SIZE [virtual_capacity=SIZE] [emergency_buffer=SIZE]
[ssd_cache=YESNO] [compression=YESNO]
```

Arguments

`emergency_buffer`	DISABLED, UNLIMITED, or a positive number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1TB, 1000000000000
`virtual_capacity`	A positive number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1TB, 1000000000000
`physical_capacity`	A positive number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1TB, 1000000000000
`name`	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
`compression`	Either yes, or no
`ssd_cache`	Either yes, or no

Creating a pool: minimal required configuration

Example

```
pool.create name=pool1 physical_capacity=20t
```

Output

```
Pool pool1 created
```

Creating an overprovisioned pool.

- The specific attributes that are relevant to the pool's physical and virtual capacity are visible through the pool.query command.

Example

```
pool.create name=pool2 physical_capacity=10t virtual_capacity=20t emergency_buffer=UNLIMITED
```

Output

```
Pool pool2 created
```

Creating an overprovisioned pool with no emergency buffer.

Example

```
pool.create name=pool3 physical_capacity=10t virtual_capacity=20t emergency_buffer=DISABLED
```

Output

```
Pool pool3 created
```

Creating a pool with an emergency buffer.

Example

```
pool.create name=pool4 physical_capacity=10t emergency_buffer=1t
```

Output

```
Emergency buffer for pool pool4 was rounded up to 932.00 GiB (1.00 TB)
Pool pool4 created
```

pool.delete ADMIN

CHANGE

Description Delete pool.

Prior to the deletion of the pool, it has to be emptied.

- All of its entities - volumes, filesystems and their snapshots - need to be moved to other pools.

Syntax

```
pool.delete pool=POOL[,POOL,...]
```

Arguments

pool	Name of an existing pool (multiple values, separated by commas)

Example

```
pool.delete name=pool1
```

Output

```
Pool pool1 deleted
```

pool.lock ADMIN

CHANGE

Description **Lock pool, revoking write access to its volumes.**

When the pool is locked, no entity can be created and volumes cannot be resized. Release the Locked state (unlock the pool) using the pool.unlock command.

Syntax

```
pool.lock pool=POOL
```

Arguments

pool	Name of an existing pool

Example

```
pool.lock name=pool1
```

Output

```
Pool pool1 Locked
```

pool.query ALL ROLES

CHANGE

Description **List existing pools.**

Run the command with no arguments in order to list all of the pools on the InfiniBox. Run the command with a pool name as an input, in order to list only this pool. Run the command with names of several pools, separated with a comma and no space, in order to list these pools. Entering a name of a pool that does not exist fails the command even if other names are entered correctly. Use the unit argument to specify the units of the capacity fields of the output.

Syntax

```
pool.query [pool=POOL[,POOL,...]] [unit=CAPACITYUNIT]
```

Arguments

pool	Name of an existing pool (multiple values, separated by commas)
unit	B, G (or GB), GiB, T (or TB), TiB or block

Example

```
pool.query
```

Output

NAME	STATE	PHYSICAL TOTAL	PHYSICAL ALLOCATED	VIRTUAL TOTAL	VIRTUAL ALLOCATED	EMERGENCY BUFFER	COMPRESSED	DATA REDUCTION
pool-1	NORMAL	3.23 TB	1.97 TB	3.23 TB	1.97 TB	DISABLED	yes	1.00:1

To set the capacity units, run the command with the unit argument.

Example

```
pool.query unit=block
```

Output

NAME	STATE	PHYSICAL TOTAL	PHYSICAL ALLOCATED	VIRTUAL TOTAL	VIRTUAL ALLOCATED	EMERGENCY BUFFER	COMPRESSED	DATA REDUCTION
pool-1	NORMAL	6314441472 blocks	3848399360 blocks	6314441472 blocks	3848399360 blocks	DISABLED	yes	1.00:1

Querying for pools and the QoS policy they are assigned to

Example

```
pool.query --columns=name,volumes_qos_policy,filesystems_qos_policy
```

Output

NAME	VOLUMES QoS POLICY	FILESYSTEMS QoS POLICY
p1	q3	q4

pool.remove_admin ADMIN

CHANGE

Description Revoke pool provisioning privileges from a pool administrator.

Syntax

```
pool.remove_admin pool=POOL user=POOLADMIN[,POOLADMIN,...]
```

Arguments

pool	Name of an existing pool
user	Name of an existing local user or ldap group, with PoolAdmin role (multiple values, separated by commas)

Example

```
pool.remove_admin name=pool1 admin=pool-admin-may-17
```

Output

```
Revoked provisioning privileges on pool pool1 from pool admins: local user pool-admin-may-17
```

pool.rename ADMIN

CHANGE

Description Rename pool.

Syntax

```
pool.rename pool=POOL new_name=NAME
```

Arguments

new_name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
pool	Name of an existing pool

Example

```
pool.rename name=pool1 new_name=pool2
```

Output

```
Pool pool1 renamed to pool2
```

pool.resize ADMIN

CHANGE

Description Resize pool.

Change the pool physical capacity, virtual capacity or emergency buffer. See the documentation of pool.create for the way the emergency buffer can be configured.

Syntax

```
pool.resize pool=POOL [physical_capacity=SIZEDELTA] [virtual_capacity=SIZEDELTA]
[emergency_buffer=SIZE]
```

Arguments

emergency_buffer	DISABLED, UNLIMITED, or a positive number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1TB, 1000000000000
virtual_capacity	A number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1TB, 1000000000000. When preceded by a plus or minus sign, represents a delta; e.g. 3000000m (absolute size), +2000GB (positive delta), -1.5T (negative delta)
physical_capacity	A number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1TB, 1000000000000. When preceded by a plus or minus sign, represents a delta; e.g. 3000000m (absolute size), +2000GB (positive delta), -1.5T (negative delta)
pool	Name of an existing pool

Resizing the pool's physical capacity.

Example

```
pool.resize name=pool1 physical_capacity=2t
```

Output

```
Pool pool1 resized
```

Resizing the pool's emergency buffer.

Example

```
pool.resize name=pool2 emergency_buffer=1t
```

Output

```
Emergency buffer for pool pool2 was rounded up to 932.00 GiB (1.00 TB)
Pool pool2 created
```

pool.set_compression ADMIN POOL ADMIN CHANGE

Description Set the pool capacity to be compressed.

Syntax

```
pool.set_compression pool=POOL compression=YESNO
```

Arguments

pool	Name of an existing pool
compression	Either yes, or no

Set the pool to compress its capacity.

Example

```
pool.set_compression name=p1 compression=yes
```

Output

```
Pool "p1" Compression was enabled
```

Set the pool to stop compressing its capacity

Example

```
pool.set_compression name=p1 compression=no
```

Output

```
Pool "p1" Compression was disabled
```

pool.set_thresholds ADMIN POOL ADMIN CHANGE

Description **Set pool capacity notification thresholds.**

Set the InfiniBox to issue events whenever the pool exceeds a capacity threshold. You may set two thresholds for each pool: Warning and Critical. The size of the Warning threshold cannot exceed the size of the Critical threshold.

Syntax

```
pool.set_thresholds pool=POOL warning=PERCENT critical=PERCENT
```

Arguments

critical	An integer between 0 and 100, optionally followed by a percent symbol
warning	An integer between 0 and 100, optionally followed by a percent symbol
pool	Name of an existing pool

Example

```
pool.set_thresholds name=pool1 warning=70% critical=80%
```

Output

```
Pool pool1 updated
```

pool.unassign_qos ADMIN CHANGE

Description **Remove pool from policy.**

Syntax

```
pool.unassign_qos pool=POOL [policy=QOSPOLICY]
```

Arguments

policy	QoS policy
pool	Name of an existing pool

Example

```
pool.unassign_qos pool=p1
```

Output

```
Pool 'p1' was unassigned from policies 'q4', 'q3'
```

pool.unlock ADMIN CHANGE

Description **Unlock pool, granting write access to its volumes.**

Syntax

```
pool.unlock pool=POOL
```

Arguments

pool	Name of an existing pool

Example

```
pool.unlock name=pool1
```

Output

```
Pool pool1 unlocked
```

pool.user_query ALL ROLES CHANGE

Description List existing pools with respective administrators.

Syntax

```
pool.user_query [pool=POOL]
```

Arguments

pool	Name of an existing pool

Example

```
pool.user_query
```

Output

NAME	ADMINS
pool1	admin, pooladmin_1
pool2	admin, pooladmin_2

Chapter 2. Volume

Volume provisioning commands

Volume commands allow you to create volumes. You can use the volume commands to map each volume to a host or a cluster, to rename and resize the volume, to restore it from a snapshot, to protect it from writing, to enable SSD cache, to set it to be thin provisioned, to move it between pools and to query the InfiniBox volumes. This category does not include commands for provisioning snapshots.

vol.assign_qos ADMIN CHANGE

Description Assign volume to policy.

Syntax

```
vol.assign_qos vol=VOL[,VOL,...] policy=QOSPOLICY
```

Arguments

policy	QoS policy
vol	Name of an existing volume (multiple values, separated by commas)

Example

```
vol.assign_qos vol=v2 policy=q1
```

Output

```
Volume 'v2' was assigned to policy 'q1'
```

vol.cache ADMIN POOL ADMIN CHANGE

Description Enable or disable SSD cache.

Set the optional parameter Recursive to Yes in order to apply the SSD cache settings to the entire volume tree.

Syntax

```
vol.cache vol=VOL ssd_cache=YESNO [recursive=YESNO]
```

Arguments

ssd_cache	Either yes, or no
vol	Name of an existing volume
recursive	Either yes, or no

Example

```
vol.cache name=vol1 ssd=yes recursive=yes
```

Output

```
Volume vol1 and all descendants SSD cache enabled.
```

vol.create

ADMIN POOL ADMIN

CHANGE

Description Create a new volume.

Create a batch of volumes by stating the requested amount of volumes using the Series parameter. The volumes names are prefixed. It is recommended to denote a new prefix for each batch. The pool parameter is not required in case there is only one pool available. The volume size can be any number that is divisible by 512 (bytes). On the pool view, the volume size is represented in a number that is divisible by 64k.

Syntax

```
vol.create name=NAME[,NAME,...] size=SIZE [pool=POOL] [thin=YESNO] [ssd_cache=YESNO]
[compression=YESNO] [series=SERIES]
```

Arguments

series	Either a range of numbers, e.g "3-7", or a single number, e.g "5", which is the same as "1-5". Zero-padding the numbers yields a zero-padded sequence. The series may contain at most 100 items, and the index of the last item may not exceed 10000000000
compression	Either yes, or no
ssd_cache	Either yes, or no
thin	Either yes, or no
pool	Name of an existing pool
size	A positive number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1TB, 1000000000000
name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@() []$=!-#{}%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped. (multiple values, separated by commas)

Creating a single volume, providing a name and specifying the volume size and the pool it will created in.

Example

```
vol.create name=vol1 size=1g pool=pool1
```

Output

```
Volume vol1 created
```

Creating a series of volumes specifying their number. The volumes will be numbered from 1 onward. Note that the name parameter serves as a prefix to the entire series.

Example

```
vol.create name=vol- size=1g pool=pool1 series=10
```

Output

```
Created 10 volumes: vol-01 through vol-10
```

Creating a series of volumes specifying the first and last ordinal numbers.

Example

```
vol.create name=vol- size=1g pool=pool1 series=11-20
```

Output

```
Created 10 volumes: vol-11 through vol-20
```

vol.delete

ADMIN POOL ADMIN

CHANGE

Description Delete volume.

You may select multiple volumes, separated with a comma.

Syntax

```
vol.delete vol=VOL[,VOL,...]
```

Arguments

vol	Name of an existing volume (multiple values, separated by commas)

Example

```
vol.delete name=vol1
```

Output

```
Volume vol1 deleted
```

vol.map

ADMIN POOL ADMIN

CHANGE

Description Map volume to host or cluster.

You may determine the LUN that will connect the volume to the host, or cluster.

Syntax

```
vol.map vol=VOL[,VOL,...] [host=HOST] [cluster=CLUSTER] [lun=LUN]
```

Arguments

cluster	Name of an existing cluster
host	Name of an existing host
vol	Name of an existing volume (multiple values, separated by commas)
lun	SCSI logical unit number (LUN)

Example

```
vol.map name=vol1 cluster=cluster1
```

Output

```
volume vol1 mapped to LUN 12 in cluster cluster1
```

vol.map_query

ALL ROLES

CHANGE

Description List host and cluster mappings by volume.

Syntax

```
vol.map_query [vol=VOL[,VOL,...]] [pool=POOL] [host=HOST] [cluster=CLUSTER]
```

Arguments

`host`	Name of an existing host
`pool`	Name of an existing pool
`vol`	Name of an existing volume (multiple values, separated by commas)
`cluster`	Name of an existing cluster

Example

```
vol.map_query
```

Output

NAME	MAPPING TYPE	MAPPED TO	LUN ID
vol1	CLUSTER	cluster1	11
vol2	CLUSTER	cluster1	12

vol.move

ADMIN POOL ADMIN

CHANGE

Description

Move volume to a different pool.

The move_capacity parameter determines whether the volume moves along with its capacity, so as not to impact the new pool free space. The default value of this parameter is no.

Syntax

```
vol.move vol=VOL pool=POOL [move_capacity=YESNO]
```

Arguments

`move_capacity`	Either yes, or no
`pool`	Name of an existing pool
`vol`	Name of an existing volume

Moving a volume with its capacity

Example

```
vol.move name=vol1 pool=pool2 move_capacity=yes
```

Output

```
Volume vol1 moved to pool pool2 with its capacity
```

Moving a volume without its capacity

Example

```
vol.move name=vol1 pool=pool2
```

Output

```
Volume vol1 moved to pool pool2
```

vol.query

ALL ROLES

CHANGE

Description

List existing volumes.

List existing volumes only.

- Use the unit argument to specify the units of the capacity fields of the output.

Syntax

```
vol.query [vol=VOL[,VOL,...]] [cg=CG] [pool=POOL] [unit=CAPACITYUNIT] [mapped=YESNO]
[write_protected=YESNO] [thin=YESNO] [compression_enabled=YESNO] [ssd_enabled=YESNO]
```

Arguments

compression_enabled	Either yes, or no
thin	Either yes, or no
write_protected	Either yes, or no
mapped	Either yes, or no
unit	B, G (or GB), GiB, T (or TB), TiB or block
pool	Name of an existing pool
cg	Name of an existing consistency group
vol	Name of an existing volume (multiple values, separated by commas)
ssd_enabled	Either yes, or no

Querying for all volumes

Example

```
vol.query
```

Output

NAME	THIN	SIZE	USED	ALLOCATED	COMPRESSION SAVINGS	SNAPSHOTS	POOL	WP	MAPPED
v1	no	1.00 TB	100 GB	1.00 TB	1.00 : 1	0	p1	no	yes

Querying for volumes that belong to a consistency group, returning the ID of the consistency group

Example

```
vol.query --columns=name,cg_id
```

Output

NAME	CG ID
v1	1272
v2	1272
v3	-

Querying for compressed volumes, and displaying only compression-related columns

Example

```
vol.query compression_enabled=yes --columns=name,compressed,data_reduction
```

Output

NAME	COMPRESSED	DATA REDUCTION
v1	yes	9.34:1
v2	yes	1.64:1
v3	yes	1.64:1

Querying for volumes that are assigned to a QoS policy

Example

```
vol.query --columns=name,policy
```

Output

NAME	POLICY
v1	q1
v2	-

vol.reclaimable ADMIN POOL ADMIN

CHANGE

Description Estimate reclaimed capacity (deletion simulation).

You may simulate the deletion of the filesystem and its descendants.

Syntax

```
vol.reclaimable vol={VOL|SNAP}
```

Arguments

vol	Name of an existing volume or snapshot

Example

```
vol.reclaimable name=vol1
```

Output

```
Estimated reclaimable space for volume v1 and its tree descendants: 327.68 KB
```

vol.rename ADMIN POOL ADMIN CHANGE

Description Rename volume.

Syntax

```
vol.rename vol=VOL new_name=NAME
```

Arguments

new_name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
vol	Name of an existing volume

Example

```
vol.rename name=vol1 new_name=vol2
```

Output

```
Volume volume1 renamed to volume2
```

vol.resize ADMIN POOL ADMIN CHANGE

Description Resize volume.

Syntax

```
vol.resize vol=VOL size=SIZEPOSITIVEDELTA
```

Arguments

size	A number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1TB, 1000000000000. When preceded by a plus sign, represents a delta; e.g. 3000000m (absolute size), +2000GB (positive delta)
vol	Name of an existing volume

Example

```
vol.resize name=vol1 size=1.5t
```

Output

```
Volume vol1 updated
```

vol.restore ADMIN POOL ADMIN CHANGE

Description **Restore data state of a volume from one of its snapshots.**

This command restores a volume from one of its snapshots.

- The volume can be restored from any of its snapshots.

Syntax

```
vol.restore vol=VOL source=SNAP
```

Arguments

source	Name of an existing volume snapshot
vol	Name of an existing volume

Example

```
vol.restore name=vol1 snap=snap1
```

Output

```
Volume vol1 restored from snapshot snap1
```

vol.set_compression ADMIN POOL ADMIN CHANGE

Description **Set a volume to use compression.**

Syntax

```
vol.set_compression vol=VOL compression=YESNO
```

Arguments

compression	Either yes, or no
vol	Name of an existing volume

Enable compression on a volume

Example

```
vol.set_compression name=v1 enable=yes
```

Output

```
Volume "v1" Compression was enabled
```

Disable compression on a volume

Example

```
vol.set_compression name=v1 enable=no
```

Output

```
Volume "v1" Compression was disabled
```

vol.set_thin ADMIN POOL ADMIN CHANGE

Description **Set volume provisioning type.**

Toggle between thin and thick provisioning.

Syntax

```
vol.set_thin vol=VOL thin=YESNO
```

Arguments

thin	Either yes, or no
vol	Name of an existing volume

Example

```
vol.set_thin name=vol1 thin=yes
```

Output

```
Volume vol1 set to thin provisioning
```

vol.tree ALL ROLES

CHANGE

Description **Display hierarchy of volumes and their snapshots.**

You may filter the results by volume and pool.

Syntax

```
vol.tree [vol=VOL] [pool=POOL] [unit=CAPACITYUNIT]
```

Arguments

unit	B, G (or GB), GiB, T (or TB), TiB or block
pool	Name of an existing pool
vol	Name of an existing volume

Example

```
vol.tree name=v1
```

Output

NAME	TYPE	SIZE	USED	ALLOCATED	POOL	WP	CREATED AT
v1	VOL	1.00 TB	100 GB	1.00 TB	p1	no	2015-12-01 10:00:00
|-----s1	SNAP	0	0	0	p1	yes	2015-12-02 10:00:00

vol.unassign_qos ADMIN

CHANGE

Description **Unassign volume from policy.**

- Unassigning the volume does not unassign its snapshots, even if both volume and snapshot are assigned to the same policy.
- This command allows for a multiple-entity selection

Syntax

```
vol.unassign_qos vol=VOL[,VOL,...]
```

Arguments

vol	Name of an existing volume (multiple values, separated by commas)

Unassigning a single volume

Example

```
vol.unassign_qos vol=v2
```

Output

```
Volume 'v2' was unassigned from policy 'q1'
```

Unassigning several volumes

Example

```
vol.unassign_qos vol=v1,v2
```

Output

```
2 volumes were unassigned from their policies
```

vol.unmap

ADMIN POOL ADMIN

CHANGE

Description Unmap volume from host or cluster.

Syntax

```
vol.unmap vol=VOL[,VOL,...] [host=HOST] [cluster=CLUSTER]
```

Arguments

cluster	Name of an existing cluster
host	Name of an existing host
vol	Name of an existing volume (multiple values, separated by commas)

Example

```
vol.unmap name=vol1 cluster=cluster1
```

Output

```
Volume vol1 unmapped from LUN LUN1 in cluster cluster1
```

vol.write_enable

ADMIN POOL ADMIN

CHANGE

Description Enable write access on volume.

Set the optional parameter Recursive to Yes in order to enable writing to the entire volume tree.

Syntax

```
vol.write_enable vol=VOL [recursive=YESNO]
```

Arguments

vol	Name of an existing volume
recursive	Either yes, or no

Example

```
vol.write_enable name=vol1 recursive=yes
```

Output

```
Volume vol1 and all descendants write access enabled
```

vol.write_protect

ADMIN POOL ADMIN

CHANGE

Description Disable write access on volume.

Set the optional parameter Recursive to Yes in order to disable writing to the entire volume tree.

Syntax

```
vol.write_protect vol=VOL [recursive=YESNO]
```

Arguments

vol	Name of an existing volume
recursive	Either yes, or no

Example

```
vol.write_protect name=vol1 recursive=yes
```

Output

```
Volume vol1 and all descendants write access disabled
```

Chapter 3. Volume Snapshot

The snapshot is an entity that allows the user to restore a volume to a specific point-in-time.

- When the snapshot is created it is in read-only mode. It can become writable (by a specific user action).
- A writable snapshot can restore its parent volume (or parent snapshot).
- Snapshot operations create a snapshot, delete it, rename it, enable SSD cache, view its relation with the volume and other snapshots, map it to a host or a cluster, unmap it, query the mapping and query all snapshots per volume or pool.

vol.snap.assign_qos ADMIN

CHANGE

Description Assign volume to policy.

Syntax

```
vol.snap.assign_qos snap=SNAP[,SNAP,...] policy=QOSPOLICY
```

Arguments

policy	QoS policy
snap	Name of an existing volume snapshot (multiple values, separated by commas)

Example

```
vol.snap.assign_qos snap=v2-s1 policy=q1
```

Output

```
Volume 'v2-s1' was assigned to policy 'q1'
```

vol.snap.cache ADMIN POOL ADMIN

CHANGE

Description Enable or disable SSD cache.

- The snapshot setting to use the InfiniBox SSD drives is inherited from the volume
- This command allows setting the snapshot to use/unuse the SSD drives independently of the volume (or parent snapshot)
- For a snapshot that has child snapshot(s) there is an option to set the SSD usage all down the snapshots tree at once (the snapshot ancestors are not affected)

Syntax

```
vol.snap.cache snap=SNAP ssd_cache=YESNO [recursive=YESNO]
```

Arguments

ssd_cache	Either yes, or no
snap	Name of an existing volume snapshot
recursive	Either yes, or no

Setting the snapshot to use SSD

Example

```
vol.snap.cache snap=v1-s1 ssd_cache=yes
```

Output

```
Volume snapshot "v1-s1" is now cached on SSD
```

Setting the snapshot - and its descendants - to use SSD

Example

```
vol.snap.cache snap=v1-s1 ssd_cache=yes recursive=yes
```

Output

```
Volume snapshot "v1-s1" and all of its descendants are now cached on SSD
```

vol.snap.create

ADMIN POOL ADMIN

CHANGE

Description Create a new volume snapshot.

- A snapshot can be from a volume or a snapshot
- The snapshot is created in a read-only state
  - The snapshot state can be changed from read-only to writable
- Restoring a volume or a snapshot from a snapshot
  - To restore the volume from its snapshot, use the vol.restore command
  - To restore the snapshot from its child snapshot, use the vol.snap.restore command

Syntax

```
vol.snap.create vol={VOL|SNAP} name=NAME [ssd_cache=YESNO]
```

Arguments

ssd_cache	Either yes, or no
name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
vol	Name of an existing volume or snapshot

Taking a snapshot of a volume

Example

```
vol.snap.create vol=vol1 snap=snap1
```

Output

```
Volume snapshot snap1 created
```

Taking a snapshot of a snapshot

Example

```
vol.snap.create vol=snap1 name=snap1-s1
```

Output

```
Volume snapshot "snap1-s1" created
```

Taking a snapshot and disabling the usage of SSD drives

Example

```
vol.snap.create vol=snap1 snap=snap1-s1 ssd_cache=no
```

Output

```
Volume snapshot "snap1-s1" created
```

vol.snap.delete

ADMIN POOL ADMIN

CHANGE

Description Delete volume snapshot.

- This is a dangerous operation that requires a specific user approval.

Syntax

```
vol.snap.delete snap=SNAP[,SNAP,...]
```

Arguments

snap	Name of an existing volume snapshot (multiple values, separated by commas)

Deleting a snapshot

Example

```
vol.snap.delete snap=v1-s3
```

Output

```
Volume snapshot "v1-s3" deleted
```

Deleting a snapshot that has children

- The dangerous operation that the user has to approve indicates that the snapshot has children

Example

```
vol.snap.delete snap=v1-s3
```

Output

```
Volume snapshot "v1-s3" deleted
```

vol.snap.map

ADMIN POOL ADMIN

CHANGE

Description Map volume snapshot to host or cluster.

- The LUN is assigned by InfiniBox

Syntax

```
vol.snap.map snap=SNAP[,SNAP,...] [host=HOST] [cluster=CLUSTER] [lun=LUN]
```

Arguments

lun	SCSI logical unit number (LUN)
cluster	Name of an existing cluster
host	Name of an existing host
snap	Name of an existing volume snapshot (multiple values, separated by commas)

Example

```
vol.snap.map snap=v1-s1-s1 host=h1
```

Output

```
Volume snapshot "v1-s1-s1" mapped to LUN 2 in host "h1"
```

vol.snap.map_query

ALL ROLES

CHANGE

Description List mappings of snapshots to hosts and clusters.

Syntax

```
vol.snap.map_query [snap=SNAP[,SNAP,...]] [pool=POOL] [host=HOST] [cluster=CLUSTER]
```

Arguments

host	Name of an existing host
pool	Name of an existing pool
snap	Name of an existing volume snapshot (multiple values, separated by commas)
cluster	Name of an existing cluster

Example

```
vol.snap.map_query snap=snap1
```

Output

NAME	MAPPING TYPE	MAPPED TO	LUN ID
snap1	CLUSTER	cluster1	11

vol.snap.query ALL ROLES

CHANGE

Description List existing snapshots.

The output of this command displays all of the snapshots on the InfiniBox system.

- Use the unit argument to specify the units of the capacity fields of the output.

Syntax

```
vol.snap.query [snap=SNAP[,SNAP,...]] [source={VOL|SNAP}] [sg=SG] [pool=POOL] [unit=CAPACITYUNIT] [mapped=YESNO] [write_protected=YESNO]
```

Arguments

unit	B, G (or GB), GiB, T (or TB), TiB or block
pool	Name of an existing pool
sg	Name of an existing snapshot group
source	Name of an existing volume or snapshot
snap	Name of an existing volume snapshot (multiple values, separated by commas)
write_protected	Either yes, or no
mapped	Either yes, or no

Querying for all of the snapshots in the system

Example

```
vol.snap.query
```

Output

NAME	THIN	SIZE	USED	ALLOCATED	POOL	WP	MAPPED	CREATED AT
v1-s1	yes	10.00 GB	0	0	p1	yes	yes	2016-04-11 10:00:00
v1-s1-s1	yes	10.00 GB	0	0	p1	yes	yes	2016-04-11 10:10:00

Querying for snapshots that are assigned to a QoS policy

Example

```
vol.snap.query --columns=name,policy
```

Output

NAME	POLICY
v1-s1	q1

vol.snap.refresh ADMIN POOL ADMIN

CHANGE

Description Refresh a snapshot.

- This operation refreshes the snapshot with the volume's data, keeping the snapshot metadata as is
- This operation is a dangerous operation

Syntax

```
vol.snap.refresh snap=SNAP [force=YESNO]
```

Arguments

`snap`	Name of an existing volume snapshot
`force`	Either yes, or no

Example

```
vol.snap.refresh snap=v1-s1
```

Output

```
Volume snapshot "v1-s1" refreshed
```

vol.snap.rename ADMIN POOL ADMIN

CHANGE

Description Rename volume snapshot.

Syntax

```
vol.snap.rename snap=SNAP new_name=NAME
```

Arguments

`new_name`	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@()[]$=!-#{}%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
`snap`	Name of an existing volume snapshot

Example

```
vol.snap.rename snap=snap1 new_name=snap2
```

Output

```
Volume snapshot snap1 renamed to snap2
```

vol.snap.resize ADMIN POOL ADMIN

CHANGE

Description Resize volume snapshot.

- The snapshot size can only be increased
- The snapshot has to be write-enabled

Syntax

```
vol.snap.resize snap=SNAP size=SIZEPOSITIVEDELTA
```

Arguments

`size`	A number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1TB, 1000000000000. When preceded by a plus sign, represents a delta; e.g. 3000000m (absolute size), +2000GB (positive delta)
`snap`	Name of an existing volume snapshot

Resizing a snapshot

Example

```
vol.snap.resize snap=s1 size=100g
```

Output

```
Volume snapshot "s1" resized
```

Running the command with an input that is identical to the snapshot current size does not fail the command

Example

```
vol.snap.resize snap=s1 size=100g
```

Output

```
Volume snapshot "s1" size is already 100.00 GB
```

vol.snap.restore

ADMIN POOL ADMIN

CHANGE

Description Restore a snapshot from one of its snapshots.

- The restored snapshot has to be in a write_enabled state
- This is a dangerous operation

Syntax

```
vol.snap.restore snap=SNAP source=SNAP
```

Arguments

source	Name of an existing volume snapshot
snap	Name of an existing volume snapshot

Example

```
vol.snap.restore target=v1-s1 source=v1-s1-s1
```

Output

```
Volume snapshot "v1-s1" restored from volume "v1-s1-s1"
```

vol.snap.tree

ALL ROLES

CHANGE

Description Display the snapshots tree.

Syntax

```
vol.snap.tree snap=SNAP [unit=CAPACITYUNIT]
```

Arguments

unit	B, G (or GB), GiB, T (or TB), TiB or block
snap	Name of an existing volume snapshot

Example

```
vol.snap.tree snap=v1-s1
```

Output

NAME	TYPE	SIZE	USED	ALLOCATED	POOL	WP	CREATED AT
v1-s1	SNAPSHOT	10.00 GB	0	0	p1	no	2016-04-11 10:00:00
| v1-s1-s1	SNAPSHOT	10.00 GB	0	0	p1	no	2016-04-11 10:00:00

vol.snap.unassign_qos

ADMIN

CHANGE

Description Unassign volume from policy.

Syntax

```
vol.snap.unassign_qos snap=SNAP[,SNAP,...]
```

Arguments

snap	Name of an existing volume snapshot (multiple values, separated by commas)

Example

```
vol.snap.unassign_qos snap=v2-s1
```

Output

```
Volume 'v2-s1' was unassigned from policy 'q1'
```

vol.snap.unmap

ADMIN POOL ADMIN

CHANGE

Description Unmap volume snapshot from host or cluster.

Syntax

```
vol.snap.unmap snap=SNAP[,SNAP,...] [host=HOST] [cluster=CLUSTER]
```

Arguments

cluster	Name of an existing cluster
host	Name of an existing host
snap	Name of an existing volume snapshot (multiple values, separated by commas)

Example

```
vol.snap.unmap snap=snap1
```

Output

```
volume snapshot snap1 unmapped from LUN 12 in cluster cluster1
```

vol.snap.write_enable

ADMIN POOL ADMIN

CHANGE

Description Enable write access on volume snapshot.

Syntax

```
vol.snap.write_enable snap=SNAP [recursive=YESNO]
```

Arguments

recursive	Either yes, or no
snap	Name of an existing volume snapshot

Example

```
vol.snap.write_enable snap=v1-s1
```

Output

```
Volume snapshot "v1-s1" is now write-enabled
```

vol.snap.write_protect

ADMIN POOL ADMIN

CHANGE

Description Disable write access on volume snapshot.

Syntax

```
vol.snap.write_protect snap=SNAP [recursive=YESNO]
```

Arguments

snap	Name of an existing volume snapshot
recursive	Either yes, or no

Example

```
vol.snap.write_protect snap=v1-s1
```

Output

```
Volume snapshot "v1-s1" is now write-protected
```

Chapter 4. Filesystem

fs.assign_qos ADMIN

CHANGE

Description Assign filesystem to policy.

Syntax

```
fs.assign_qos fs=FS[,FS,...] policy=QOSPOLICY
```

Arguments

policy	QoS policy
fs	Name of an existing filesystem (multiple values, separated by commas)

Example

```
fs.assign_qos fs=fs1 policy=q2
```

Output

```
Filesystem 'fs1' was assigned to policy 'q2'
```

fs.cache ADMIN POOL ADMIN

CHANGE

Description Enable or disable SSD cache.

Set the optional parameter Recursive to Yes in order to apply the SSD cache settings to the entire volume tree.

Syntax

```
fs.cache fs=FS ssd_cache=YESNO [recursive=YESNO]
```

Arguments

fs	Name of an existing filesystem
recursive	Either yes, or no
ssd_cache	Either yes, or no

Example

```
fs.cache name=fs1 ssd=yes recursive=yes
```

Output

```
Filesystem fs1 and all descendants are SSD cache enabled
```

fs.create ADMIN POOL ADMIN

CHANGE

Description Create a new filesystem.

Syntax

```
fs.create name=NAME[,NAME,...] size=SIZE [pool=POOL] [thin=YESNO] [ssd_cache=YESNO] [series=SERIES]
[compression=YESNO]
```

Arguments

`size`	A positive number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1TB, 1000000000000
`name`	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped. (multiple values, separated by commas)
`compression`	Either yes, or no
`series`	Either a range of numbers, e.g "3-7", or a single number, e.g "5", which is the same as "1-5". Zero-padding the numbers yields a zero-padded sequence. The series may contain at most 100 items, and the index of the last item may not exceed 10000000000
`ssd_cache`	Either yes, or no
`thin`	Either yes, or no
`pool`	Name of an existing pool

Creating a single filesystem, providing a name and specifying its size and the pool it will created in.

Example

```
fs.create name=fs1 size=1g pool=pool1
```

Output

```
Volume fs1 created
```

Creating a series of filesystems specifying their number. The filesystems will be numbered from 1 onward. Note that the name parameter serves as a prefix to the entire series.

Example

```
fs.create name=fs- size=1g pool=pool1 series=10
```

Output

```
Created 10 filessytems: fs-01 through fs-10
```

Creating a series of filesystems specifying the first and last ordinal numbers.

Example

```
fs.create name=fs- size=1g pool=pool1 count=11-20
```

Output

```
Created 10 filesystems: fs-11 through fs-20
```

fs.delete ADMIN POOL ADMIN CHANGE

Description Delete filesystem.

This command succeeds only if the filesystem has no active exports.

Syntax

```
fs.delete fs=FS[,FS,...]
```

Arguments

`fs`	Name of an existing filesystem (multiple values, separated by commas)

Example

```
fs.delete name=fs1
```

Output

```
Filesystem fs1 deleted
```

fs.move ADMIN POOL ADMIN

CHANGE

Description Move filesystem to a different pool.

Syntax

```
fs.move fs=FS pool=POOL [move_capacity=YESNO]
```

Arguments

move_capacity	Either yes, or no
pool	Name of an existing pool
fs	Name of an existing filesystem

Moving a filesystem with its capacity

Example

```
fs.move name=fs1 pool=pool2 move_capacity=yes
```

Output

```
Filesystem fs1 moved to pool pool2 without its capacity
```

Moving a filesystem without its capacity

Example

```
fs.move name=fs1 pool=pool2
```

Output

```
Filesystem fs1 moved to pool pool2
```

fs.query ALL ROLES

CHANGE

Description List existing filesystems.

Syntax

```
fs.query [fs=FS[,FS,...]] [pool=POOL] [unit=NASCAPACITYUNIT] [has_exports=YESNO]
[write_protected=YESNO] [thin=YESNO] [compression_enabled=YESNO] [ssd_enabled=YESNO]
```

Arguments

thin	Either yes, or no
write_protected	Either yes, or no
has_exports	Either yes, or no
unit	B, G (or GB), GiB, T (or TB), TiB
pool	Name of an existing pool
fs	Name of an existing filesystem (multiple values, separated by commas)
compression_enabled	Either yes, or no
ssd_enabled	Either yes, or no

Example

```
fs.query name=fs1
```

Output

NAME	THIN	SIZE	USED	ALLOCATED	COMPRESSION SAVINGS	SNAPSHOTS	POOL	WP	EXPORTED	COMPRESSION
fs1	yes	1.00 GB	327.68 KB	327.68 KB	1.00 : 1	0	p1	no	yes	yes

Querying for filesystems that are assigned to a QoS policy

Example

```
fs.query --columns=name,policy
```

Output

NAME	POLICY
fs1	q2

fs.reclaimable

ADMIN POOL ADMIN

CHANGE

Description Estimate reclaimed capacity (deletion simulation).

You may simulate the deletion of the filesystem and its descendants.

Syntax

```
fs.reclaimable fs={FS|SNAP}
```

Arguments

fs	Name of an existing filesystem or snapshot

Example

```
fs.reclaimable name=fs1
```

Output

```
Estimated reclaimable space for filesystem fs1 and its tree descendants: 1.00 TB
```

fs.rename

ADMIN POOL ADMIN

CHANGE

Description Rename filesystem.

Syntax

```
fs.rename fs=FS new_name=NAME
```

Arguments

new_name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@()[]$=!-#{}%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
fs	Name of an existing filesystem

Example

```
fs.rename name=fs1 new_name=fs2
```

Output

```
Filesystem fs1 renamed to fs2
```

fs.resize

ADMIN POOL ADMIN

CHANGE

Description Resize filesystem.

The filesystem can only be resized upwards. Its size cannot be decreased.

Syntax

```
fs.resize fs=FS size=SIZEPOSITIVEDELTA
```

Arguments

size	A number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1TB, 1000000000000. When preceded by a plus sign, represents a delta; e.g. 3000000m (absolute size), +2000GB (positive delta)
fs	Name of an existing filesystem

Example

```
fs.resize name=fs1 size=1GB
```

Output

```
Filesystem fs1 resized
```

fs.restore ADMIN POOL ADMIN

CHANGE

Description This command restores a filesystem from one of its snapshots.

Syntax

```
fs.restore fs=FS source=SNAP
```

Arguments

source	Name of an existing filesystem snapshot
fs	Name of an existing filesystem

Example

```
fs.restore name=fs1 snap=fs1s1
```

Output

```
Filesystem fs1 restored from snapshot fs1s1
```

fs.set_compression ADMIN POOL ADMIN

CHANGE

Description Set the filesystem to use compression.

Syntax

```
fs.set_compression fs=FS compression=YESNO
```

Arguments

fs	Name of an existing filesystem
compression	Either yes, or no

Enable compression on a filesystem

Example

```
fs.set_compression name=fs1 enable=yes
```

Output

```
Filesystem "fs1" Compression was enabled
```

Disable compression on a filesystem

Example

```
fs.set_compression name=fs1 enable=no
```

Output

```
Filesystem "fs1" Compression was disabled
```

fs.set_thin ADMIN POOL ADMIN

CHANGE

Description Set filesystem provisioning type.

Syntax

```
fs.set_thin fs=FS thin=YESNO
```

Arguments

thin	Either yes, or no
fs	Name of an existing filesystem

Example

```
fs.set_thin name=fs1 thin=yes
```

Output

```
Filesystem fs1 set to thin provisioning
```

fs.tree ALL ROLES

CHANGE

Description Display hierarchy of filesystems along with their snapshots.

Syntax

```
fs.tree [fs=FS] [pool=POOL] [unit=NASCAPACITYUNIT]
```

Arguments

unit	B, G (or GB), GiB, T (or TB), TiB
pool	Name of an existing pool
fs	Name of an existing filesystem

Example

```
fs.tree name=fs1
```

Output

NAME	TYPE	SIZE	USED	ALLOCATED	POOL	WP	CREATED AT
fs1	FILESYSTEM	1.00 TB	100 GB	1.00 TB	p1	no	2015-12-01 10:00:00
|-----s1	SNAP	0	0	0	p1	yes	2015-12-02 10:00:00

fs.unassign_qos ADMIN

CHANGE

Description Unassign fs from policy.

- Unassigning the filesystem does not unassign its snapshots, even if both filesystem and snapshot are assigned to the same policy.
- This command allows for a multiple-entity selection

Syntax

```
fs.unassign_qos fs=FS[,FS,...]
```

Arguments

fs	Name of an existing filesystem (multiple values, separated by commas)

Unassigning a single filesystem

Example

```
fs.unassign_qos fs=fs1
```

Output

```
Filesystem 'fs1' was unassigned from policy 'q2'
```

Unassigning multiple filesystems

Example

```
fs.unassign_qos fs=fs1,fs2
```

Output

```
2 filesystems were unassigned from their policies
```

fs.write_enable ADMIN POOL ADMIN

CHANGE

Description Enable write access on filesystem.

Syntax

```
fs.write_enable fs=FS [recursive=YESNO]
```

Arguments

recursive	Either yes, or no
fs	Name of an existing filesystem

Example

```
fs.write_enable name=fs1 recursive=yes
```

Output

```
Filesystem fs1 and all descendants write access enabled
```

fs.write_protect ADMIN POOL ADMIN

CHANGE

Description Disable write access on filesystem.

Syntax

```
fs.write_protect fs=FS [recursive=YESNO]
```

Arguments

fs	Name of an existing filesystem
recursive	Either yes, or no

Example

```
fs.write_protect name=fs1 recursive=yes
```

Output

```
Filesystem fs1 and all descendants write access disabled
```

Chapter 5. Filesystem Snapshot

fs.snap.assign_qos ADMIN

CHANGE

Description Assign filesystem to policy.

Syntax

```
fs.snap.assign_qos snap=SNAP[,SNAP,...] policy=QOSPOLICY
```

Arguments

policy	QoS policy
snap	Name of an existing filesystem snapshot (multiple values, separated by commas)

Example

```
fs.snap.assign_qos snap=fs1-s1 policy=q2
```

Output

```
Filesystem 'fs1-s1' was assigned to policy 'q2'
```

fs.snap.cache ADMIN POOL ADMIN

CHANGE

Description Enable or disable SSD cache.

Use the recursive argument in order to enable SSD for the snapshot and its descendants.

Syntax

```
fs.snap.cache snap=SNAP ssd_cache=YESNO [recursive=YESNO]
```

Arguments

snap	Name of an existing filesystem snapshot
recursive	Either yes, or no
ssd_cache	Either yes, or no

Example

```
fs.snap.cache name=fs1s1 ssd=yes
```

Output

```
Filesystem snapshot fs1s1 SSD cache enabled
```

fs.snap.create ADMIN POOL ADMIN

CHANGE

Description Create a new filesystem snapshot.

Create a snapshot from a filesystem.

Syntax

```
fs.snap.create fs={FS|SNAP} name=NAME [ssd_cache=YESNO]
```

Arguments

ssd_cache	Either yes, or no
name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@() []$=!-#{}%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
fs	Name of an existing filesystem or snapshot

Example

```
fs.snap.create source=fs1 name=fs1s1
```

Output

```
Filesystem snapshot fs1s1 created
```

fs.snap.delete ADMIN POOL ADMIN

CHANGE

Description Delete filesystem snapshot.

Syntax

```
fs.snap.delete snap=SNAP[,SNAP,...]
```

Arguments

snap	Name of an existing filesystem snapshot (multiple values, separated by commas)

Example

```
fs.snap.delete name=fs1s1
```

Output

```
Filesystem snapshot fs1s1 deleted
```

fs.snap.query ALL ROLES

CHANGE

Description List existing snapshots.

This command lists either all filsystem snashots, or snapshots per specific filesystem. The source parameter of this command is the filesystem whose snapshots are queried.

Syntax

```
fs.snap.query [snap=SNAP[,SNAP,...]] [source={FS|SNAP}] [pool=POOL] [unit=NASCAPACITYUNIT]
[has_exports=YESNO] [write_protected=YESNO]
```

Arguments

snap	Name of an existing filesystem snapshot (multiple values, separated by commas)
has_exports	Either yes, or no
unit	B, G (or GB), GiB, T (or TB), TiB
pool	Name of an existing pool
source	Name of an existing filesystem or snapshot
write_protected	Either yes, or no

Example

```
fs.snap.query source=fs1
```

Output

NAME	THIN	SIZE	USED	ALLOCATED	POOL	CREATED AT
fs1s1	yes	1.00 GB	327.68 KB	65.53 KB	pool1	2015-06-01 10:00:00
fs1s2	yes	1.00 GB	327.68 KB	65.53 KB	pool1	2015-06-01 11:00:00

Querying for filesystem snapshots that are assigned to a QoS policy

Example

```
fs.snap.query --columns=name,policy
```

Output

```
[
    [
        "NAME",
        "POLICY"
    ],
    [
        "fs1",
        "q2"
    ]
]
```

fs.snap.refresh ADMIN POOL ADMIN CHANGE

Description Refresh a snapshot.

This operation refreshes the snapshot with the filesystem's data, keeping the snapshot metadata as is This operation is a dangerous operation

Syntax

```
fs.snap.refresh snap=SNAP
```

Arguments

snap	Name of an existing filesystem snapshot

Example

```
fs.snap.refresh snap=fs1-s1
```

Output

```
Filesystem snapshot "fs1-s1" refreshed
```

fs.snap.rename ADMIN POOL ADMIN

CHANGE

Description Rename filesystem snapshot.

Syntax

```
fs.snap.rename snap=SNAP new_name=NAME
```

Arguments

new_name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
snap	Name of an existing filesystem snapshot

Example

```
fs.snap.rename name=fs1s1 new_name=fs1s2
```

Output

```
Filesystem snapshot fs1s1 renamed to fs1s2
```

fs.snap.resize ADMIN POOL ADMIN

CHANGE

Description Resize filesystem snapshot.

- The snapshot size can only be increased
- The snapshot has to be write-enabled

Syntax

```
fs.snap.resize snap=SNAP size=SIZEPOSITIVEDELTA
```

Arguments

size	A number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1TB, 1000000000000. When preceded by a plus sign, represents a delta; e.g. 3000000m (absolute size), +2000GB (positive delta)
snap	Name of an existing filesystem snapshot

Resizing the snapshot

Example

```
fs.snap.resize snap=s1 size=100g
```

Output

```
Filesystem snapshot "s1" resized
```

Running the command with an input that is identical to the snapshot current size does not fail the command

Example

```
fs.snap.resize snap=s1 size=100g
```

Output

```
Filesystem snapshot "s1" size is already 100.00 GB
```

fs.snap.restore ADMIN POOL ADMIN

CHANGE

Description Restore a filesystem snapshot from one of its snapshots.

- The restored snapshot has to be in a write_enabled state
- This is a dangerous operation

Syntax

```
fs.snap.restore snap=SNAP source=SNAP
```

Arguments

snap	Name of an existing filesystem snapshot
source	Name of an existing filesystem snapshot

Example

```
fs.snap.restore target=fs1-s1 source=fs1-s1-s1
```

Output

```
Filesystem snapshot "fs1-s1" restored from snapshot "fs1-s1-s1"
```

fs.snap.tree ALL ROLES

CHANGE

Description Display hierarchy of snapshots and their snapshots.

Syntax

```
fs.snap.tree snap=SNAP [unit=NASCAPACITYUNIT]
```

Arguments

unit	B, G (or GB), GiB, T (or TB), TiB
snap	Name of an existing filesystem snapshot

Example

```
fs.snap.tree name=fs1s1
```

Output

NAME	TYPE	SIZE
fs1s1	SNAPSHOT	0.002 TB
|----fs1s1s1	SNAPSHOT	0.002 TB

fs.snap.unassign_qos ADMIN

CHANGE

Description Unassign fs from policy.

Syntax

```
fs.snap.unassign_qos snap=SNAP[,SNAP,...]
```

Arguments

snap	Name of an existing filesystem snapshot (multiple values, separated by commas)

Example

```
fs.snap.unassign_qos snap=fs1-s1
```

Output

```
Filesystem 'fs1-s1' was unassigned from policy 'q2'
```

fs.snap.write_enable ADMIN POOL ADMIN

CHANGE

Description Enable write access on filesystem snapshot.

Syntax

```
fs.snap.write_enable snap=SNAP [recursive=YESNO]
```

Arguments

recursive	Either yes, or no
snap	Name of an existing filesystem snapshot

Example

```
fs.snap.write_enable fs=fs1-s1
```

Output

```
Filesystem snapshot "fs1-s1" is now write-enabled
```

fs.snap.write_protect ADMIN POOL ADMIN CHANGE

Description Disable write access on filesystem snapshot.

Syntax

```
fs.snap.write_protect snap=SNAP [recursive=YESNO]
```

Arguments

recursive	Either yes, or no
snap	Name of an existing filesystem snapshot

Example

```
fs.snap.write_protect fs=fs1-s1
```

Output

```
Filesystem snapshot "fs1-s1" is already write-protected
```

Chapter 6. Filesystem Export

fs.export.create ADMIN POOL ADMIN CHANGE

Description Create a new filesystem export.

The following parameters accept values up to 9*1024*1024*1024 byte: maximum_write_size, maximum_read_size, preferred_write_size, preferred_readdir_size.

Syntax

```
fs.export.create fs={FS|SNAP} export_path=STR [internal_path=STR] [copy_permissions_from=EXPORT]
[squash_all_users=YESNO] [anonymous_gid=INT] [anonymous_uid=INT] [maximum_write_size=SIZE]
[maximum_read_size=SIZE] [preferred_write_size=SIZE] [preferred_read_size=SIZE]
[preferred_readdir_size=SIZE] [privileged_port=YESNO]
```

Arguments

preferred_readdir_size	A positive number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1TB, 1000000000000
preferred_read_size	A positive number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1TB, 1000000000000
preferred_write_size	A positive number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1TB, 1000000000000
maximum_read_size	A positive number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1TB, 1000000000000
maximum_write_size	A positive number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1TB, 1000000000000
anonymous_uid	An integer number
anonymous_gid	An integer number
squash_all_users	Either yes, or no
copy_permissions_from	Name of an existing export
internal_path	Any sequence of characters
export_path	Any sequence of characters
fs	Name of an existing filesystem or snapshot
privileged_port	Either yes, or no

Example

```
fs.export.create fs=fs1 export_path=/fs/
```

Output

```
Export /fs/ created
```

fs.export.delete ADMIN POOL ADMIN

CHANGE

Description Delete filesystem export.

Syntax

```
fs.export.delete export_path=EXPORT[,EXPORT,...]
```

Arguments

export_path	Name of an existing export (multiple values, separated by commas)

Example

```
fs.export.delete export_path=/fs/
```

Output

```
Export /fs/ deleted
```

fs.export.disable ADMIN POOL ADMIN

CHANGE

Description Disable filesystem export.

Syntax

```
fs.export.disable export_path=EXPORT
```

Arguments

export_path	Name of an existing export

Example

```
fs.export.disable export_path=/fs/
```

Output

```
Export /fs/ disabled
```

fs.export.enable

ADMIN POOL ADMIN

CHANGE

Description Enable filesystem export.

Syntax

```
fs.export.enable export_path=EXPORT
```

Arguments

export_path	Name of an existing export

Example

```
fs.export.enable export_path=/fs/
```

Output

```
Export /fs/ enabled
```

fs.export.modify

ADMIN POOL ADMIN

CHANGE

Description Modify existing filesystem export.

The following parameters accept values up to 9*1024*1024*1024 byte: maximum_write_size, maximum_read_size, preferred_write_size, preferred_readdir_size.

Syntax

```
fs.export.modify export_path=EXPORT [copy_permissions_from=EXPORT] [squash_all_users=YESNO]
[anonymous_gid=INT] [anonymous_uid=INT] [maximum_write_size=SIZE] [maximum_read_size=SIZE]
[preferred_write_size=SIZE] [preferred_read_size=SIZE] [preferred_readdir_size=SIZE]
[privileged_port=YESNO]
```

Arguments

privileged_port	Either yes, or no
preferred_readdir_size	A positive number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1TB, 1000000000000
preferred_read_size	A positive number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1TB, 1000000000000
preferred_write_size	A positive number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1TB, 1000000000000
maximum_read_size	A positive number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1TB, 1000000000000
maximum_write_size	A positive number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1TB, 1000000000000
anonymous_uid	An integer number
anonymous_gid	An integer number
squash_all_users	Either yes, or no
copy_permissions_from	Name of an existing export
export_path	Name of an existing export

Example

```
fs.export.modify export_path=/fs/ copy_permissions_from=/fs1/
```

Output

```
Export /fs/ modified
```

fs.export.query ALL ROLES

CHANGE

Description List existing filesystem exports.

Syntax

```
fs.export.query [export_path=EXPORT[,EXPORT,...]] [internal_path=STR[,STR,...]] [fs={FS|SNAP}[,{FS|SNAP},...]]
```

Arguments

fs	Name of an existing filesystem or snapshot (multiple values, separated by commas)
internal_path	Any sequence of characters (multiple values, separated by commas)
export_path	Name of an existing export (multiple values, separated by commas)

Running the command for a filesystem.

Example

```
fs.export.query name=FS1010
```

Output

EXPORT PATH	INTERNAL PATH	FILESYSTEM	TYPE	ENABLED	PERMISSIONS
/export/FS1010_a	/	FS1010	FILESYSTEM	yes	1
/export/FS1010_b	/	FS1010	FILESYSTEM	yes	1

Running the command for the same fielsystem, with detailed output.

Example

```
fs.export.query name=FS1010 --detailed
```

Output

EXPORT PATH	/export/FS1010_a
Internal Path	/
Filesystem	FS1010
Type	Filesystem
Enabled	yes
Anongid	65534
Anonuid	65534
Squash All Users	no
Max Wsize	1048576
Max Rsize	1048576
Pref Wsize	65536
Pref Rsize	65536
Pref Readdir Size	32768
Privileged Ports Only	no
32Bit File ID	no
EXPORT PATH	/export/FS1010_b
Internal Path	/
Filesystem	FS1010
Type	Filesystem
Enabled	yes
Anongid	65534
Anonuid	65534
Squash All Users	no
Max Wsize	1048576
Max Rsize	1048576
Pref Wsize	65536
Pref Rsize	65536
Pref Readdir Size	32768
Privileged Ports Only	no
32Bit File ID	no

Chapter 7. Filesystem Export Permission

fs.export.permission.add ADMIN POOL ADMIN

CHANGE

Description **Grant a host with permissions to access an existing filesystem export.**

The client can be either a host that is defined in the system, or an IP address.

Syntax

```
fs.export.permission.add export_path=EXPORT client=CLIENT access=EXPORTACCESSLEVEL
[no_root_squash=YESNO]
```

Arguments

no_root_squash	Either yes, or no
access	RO or RW
client	A full wildcard (i.e: *), IP address or IP range (e.g 10.0.0.1-10.0.0.10)
export_path	Name of an existing export

Example

```
fs.export.permission.add export_path=/fs/ client=172.16.66.172 access=READ_WRITE
```

Output

```
READ_WRITE access granted to client 172.16.66.172 on /fs/
```

fs.export.permission.modify

ADMIN POOL ADMIN

CHANGE

Description **Add a new permission to an existing filesystem export.**

This command changes the access rights of an existing client. This command does not allow to change the client. To add a client, use the fs.export.permissions.add command.

Syntax

```
fs.export.permission.modify export_path=EXPORT client=CLIENT [access=EXPORTACCESSLEVEL]
[no_root_squash=YESNO]
```

Arguments

export_path	Name of an existing export
no_root_squash	Either yes, or no
access	RO or RW
client	A full wildcard (i.e: *), IP address or IP range (e.g 10.0.0.1-10.0.0.10)

Example

```
fs.export.permission.modify export_path=/fs/ client=172.16.66.172 access=READ_ONLY
```

Output

```
READ_ONLY access granted to client 172.16.66.172 on /fs/
```

fs.export.permission.query

ALL ROLES

CHANGE

Description **List permissions for existing filesystem exports.**

Syntax

```
fs.export.permission.query [export_path=EXPORT[,EXPORT,...]] [fs={FS|SNAP}[,{FS|SNAP},...]]
[client=CLIENT[,CLIENT,...]] [access=EXPORTACCESSLEVEL[,EXPORTACCESSLEVEL,...]]
[no_root_squash=YESNO]
```

Arguments

access	RO or RW (multiple values, separated by commas)
client	A full wildcard (i.e: *), IP address or IP range (e.g 10.0.0.1-10.0.0.10) (multiple values, separated by commas)
fs	Name of an existing filesystem or snapshot (multiple values, separated by commas)
export_path	Name of an existing export (multiple values, separated by commas)
no_root_squash	Either yes, or no

Example

```
fs.export.permission.query export_path=/fs/
```

Output

EXPORT PATH	CLIENT	ACCESS LEVEL
/fs/	172.16.66.172	READ_WRITE
/fs/	172.16.66.173	READ_ONLY

fs.export.permission.remove

ADMIN POOL ADMIN

CHANGE

Description **Remove a permission from an existing filesystem export.**

Enter pairs of export path and client.

Syntax

```
fs.export.permission.remove export_path=EXPORT client=CLIENT
```

Arguments

client	A full wildcard (i.e: *), IP address or IP range (e.g 10.0.0.1-10.0.0.10)
export_path	Name of an existing export

Example

```
fs.export.permission.remove export_path=/fs/ client=172.16.66.172
```

Output

```
READ_WRITE access revoked from client 172.16.66.172 on /fs/
```

Chapter 8. Consistency group

Consistency group commands allow to group datasets together and take snapshots for all of the members at the same point-in-time.

cg.add_member

ADMIN POOL ADMIN

CHANGE

Description **Add a member to a consistency group.**

This command adds a member to a consistency group.

- A snapshot group that is taken after the member addition will include a snapshot for this new member
- Any snapshot group that was taken prior to the member addition does not include a snapshot for this new member, thus is considered inconsistent with the consistency group

Replicated consistency group

- Adding a non-replicating volume to a replicating consistency group breaks its replication consistency.
- In such a case, the consistency group replica on the target will be considered consistent as of the next sync job.
- Adding a dataset other than volume is impossible.

Syntax

```
cg.add_member cg=CG member=VOL[,VOL,...] [target=REMOTEENTITY[,REMOTEENTITY,...]]
[new_target_name=NAME[,NAME,...]]
```

Arguments

new_target_name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped. (multiple values, separated by commas)
target	Name of a remote entity (auto-completion unavailable) (multiple values, separated by commas)
member	Name of an existing volume (multiple values, separated by commas)
cg	Name of an existing consistency group

In this example, a single volume is added to a consistency group.

Example

```
cg.add_member name=cg1 members=vol1
```

Output

```
Volume vol1 added to consistency group cg1
```

In this example, three volumes are added to a consistency group.

Example

```
cg.add_member name=cg1 members=vol1,vol2,vol3
```

Output

```
3 members added to consistency group cg1
```

cg.create ADMIN POOL ADMIN CHANGE

Description Create a new consistency group.

Syntax

```
cg.create name=NAME [pool=POOL]
```

Arguments

pool	Name of an existing pool
name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.

Example

```
cg.create name=cg1
```

Output

```
Consistency group cg1 created
```

cg.delete ADMIN POOL ADMIN CHANGE

Description Delete a consistency group.

Syntax

```
cg.delete cg=CG[,CG,...] [delete_members=YESNO]
```

Arguments

delete_members	Either yes, or no
cg	Name of an existing consistency group (multiple values, separated by commas)

Example

```
cg.delete name=cg1
```

Output

```
Consistency group cg1 deleted
```

cg.member_query ALL ROLES

CHANGE

Description List all cg members.

Syntax

```
cg.member_query [cg=CG[,CG,...]]
```

Arguments

cg	Name of an existing consistency group (multiple values, separated by commas)

Example

```
cg.member_query name=cg1
```

Output

NAME	SIZE	USED	ALLOCATED	MAPPED
vol-01	10.00 TB	9.83 TB	9.83 TB	yes
vol-02	10.00 TB	9.83 TB	9.83 TB	yes
vol-03	10.00 TB	9.83 TB	9.83 TB	yes

cg.move ADMIN POOL ADMIN

CHANGE

Description Move a consistency group to another pool.

Move a consistency group from one pool or another with or without its members. See the examples below. The association of the members with a pool can be verified using the vol.query command.

Syntax

```
cg.move cg=CG pool=POOL [move_capacity=YESNO]
```

Arguments

pool	Name of an existing pool
cg	Name of an existing consistency group
move_capacity	Either yes, or no

In this example, the consistency group - and its volumes - move to another pool without their capacity.

Example

```
cg.move name=cg1 pool=pool2
```

Output

```
Consistency group cg1 moved to pool pool2 without its capacity
```

In this example, the consistency group - and its volumes - move to another pool with their capacity.

Example

```
cg.move name=cg1 pool=pool1 move_capacity=yes
```

Output

```
Consistency group cg1 moved to pool pool1 along with its capacity
```

This example is similar to the first. In this example, the consistency group - and its volumes - move to another pool without their capacity.

Example

```
cg.move name=cg1 pool=pool2 move_capacity=no
```

Output

```
Consistency group cg1 moved to pool pool2 without its capacity
```

cg.query ALL ROLES

CHANGE

Description **List existing consistency groups.**

Query consistency groups by name, the pool they belong to, or dataset members.

Syntax

```
cg.query [cg=CG[,CG,...]] [pool=POOL] [member=VOL]
```

Arguments

cg	Name of an existing consistency group (multiple values, separated by commas)
pool	Name of an existing pool
member	Name of an existing volume

Example

```
cg.query
```

Output

NAME	POOL	MEMBERS	REPLICATED
cg1	pool1	1	yes
cg2	pool2	2	no

cg.remove_member ADMIN POOL ADMIN

CHANGE

Description **Remove a member from a consistency group.**

This command removes a member from a consistency group.

- A snapshot group that is taken after the member removal will not include a snapshot for the removed member
- Any snapshot group that was taken prior to the member removal is no longer consistent with the consistency group

For a replicating consistency group, the removed member can keep being replicated

- Use the force flag to keep the member to replicate to the already existing target
- User the create_new_replica flag to create a new replica
- Use retain_staging_area (available for asynchronous replica only) to stop replicating the member, but to keep the target

Syntax

```
cg.remove_member cg=CG member=VOL [retain_staging_area=YESNO] [force=YESNO]
[create_new_replica=YESNO]
```

Arguments

create_new_replica	Either yes, or no
force	Either yes, or no
retain_staging_area	Either yes, or no
member	Name of an existing volume
cg	Name of an existing consistency group

In this example, a single volume is removed from a consistency group.

Example

```
cg.remove_member name=cg1 members=vol1
```

Output

```
Volume vol1 removed from consistency group cg1
```

In this example, a three volumes were removed from a consistency group.

Example

```
cg.add_member name=cg1 members=vol1,vol2,vol3
```

Output

```
3 members removed from consistency group cg1
```

cg.rename

ADMIN POOL ADMIN

CHANGE

Description Rename a consistency group.

Syntax

```
cg.rename cg=CG new_name=NAME
```

Arguments

new_name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@()[]$=!-#{}%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
cg	Name of an existing consistency group

Example

```
cg.rename name=cg1 new_name=cg2
```

Output

```
Consistency group cg1 renamed to cg2
```

cg.restore

ADMIN POOL ADMIN

CHANGE

Description Restore a consistency group from one of its snapshot groups.

Syntax

```
cg.restore cg=CG source=SG
```

Arguments

cg	Name of an existing consistency group
source	Name of an existing snapshot group

Example

```
cg.restore name=cg1 source=sg1
```

Output

```
Consistency group cg1 restored from snapshot group sg1
```

Chapter 9. Snapshot group

Snapshot Groups commands allow to create a group of snapshot per consistency group. The members of the snapshot group are snapshots, taken one per each memeber of the consistency group.

cg.snap.create

ADMIN POOL ADMIN

CHANGE

Description Create a snapshot of a consistency group.

The snapshot group includes a snapshot group entity and individual snapshots for each of the datasets of the consistency gorup.

- Provide the name of the consistency group, a name for the new snapshot group, and either a prefix or a suffix for the individual snapshots.
- As the maximum length of a consistency group is 65 characters, the length of the prefix, or suffix, has to take into consideration the length of the longest consistency group member

Syntax

```
cg.snap.create cg=CG name=NAME [snap_prefix=NAME] [snap_suffix=NAME]
```

Arguments

snap_suffix	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@() []$=!-#{}%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
snap_prefix	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@() []$=!-#{}%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@() []$=!-#{}%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
cg	Name of an existing consistency group

Example

```
cg.snap.create sg=sg1 source=cg1 snap_suffix=pre-initialization
```

Output

```
Snapshot group sg1 created
```

cg.snap.delete ADMIN POOL ADMIN CHANGE

Description Delete a snapshot group.

This command deletes a snapshot group with or without its members. An approval is required for this operation.

Syntax

```
cg.snap.delete sg=SG[,SG,...] [delete_members=YESNO]
```

Arguments

sg	Name of an existing snapshot group (multiple values, separated by commas)
delete_members	Either yes, or no

In this example, a snapshot group is deleted and its snapshots remain.

Example

```
cg.snap.delete sg=sg1
```

Output

```
Snapshot group sg1 deleted
```

In this example, a snapshot group is deleted and its snapshots are deleted as well.

Example

```
cg.snap.delete sg=sg1 delete_members=yes
```

Output

```
Snapshot group sg1 deleted
```

cg.snap.member_query ALL ROLES

CHANGE

Description List all members of a snapshot group.

Syntax

```
cg.snap.member_query [sg=SG[,SG,...]]
```

Arguments

sg	Name of an existing snapshot group (multiple values, separated by commas)

Example

```
cg.snap.member_query sg=sg1
```

Output

NAME	DATASET TYPE	THIN	SIZE
snap-1	SNAPSHOT	yes	1.00 GB
snap-2	SNAPSHOT	yes	1.00 GB
snap-3	SNAPSHOT	yes	1.00 GB

cg.snap.query ALL ROLES

CHANGE

Description List existing snapshot groups.

Syntax

```
cg.snap.query [sg=SG[,SG,...]] [source=CG] [pool=POOL] [member=SNAP]
```

Arguments

sg	Name of an existing snapshot group (multiple values, separated by commas)
pool	Name of an existing pool
source	Name of an existing consistency group
member	Name of an existing volume snapshot

Example

```
cg.snap.query
```

Output

NAME	POOL	MEMBERS
sg1	pool1	1
sg2	pool2	2

cg.snap.refresh ADMIN POOL ADMIN

CHANGE

Description This command refreshes all of the members of a snapshot group from the members of the consistency group.

- This command is a dangerous operation

Syntax

```
cg.snap.refresh sg=SG
```

Arguments

sg	Name of an existing snapshot group

Example

```
cg.snap.refresh sg=cg1-sg1
```

Output

```
Snapshot group "cg1-sg1" refreshed
```

cg.snap.rename ADMIN POOL ADMIN CHANGE

Description Rename a snapshot group.

Syntax

```
cg.snap.rename sg=SG new_name=NAME
```

Arguments

new_name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
sg	Name of an existing snapshot group

Example

```
cg.snap.rename sg=sg1 new_name=sg2
```

Output

```
Snapshot group sg1 renamed to sg2
```

Chapter 10. Host

The following host commands allow you to create a host, to map a volume or a snapshot to a host and to query the mapping, to add a port to a host and to query the ports by host.

host.add_port ADMIN POOL ADMIN CHANGE

Description Add port to host.

Syntax

```
host.add_port host=HOST port=SCSIINITIATOR[,SCSIINITIATOR,...] [type=TYPE]
```

Arguments

type	iSCSI or FC
port	Logged in initiator (multiple values, separated by commas)
host	Name of an existing host

Example

```
host.add_port name=host1 port=1000000c9c06f340
```

Output

```
FC port 1000000c9c06f340 added to host host1
```

host.clear_chap ADMIN POOL ADMIN CHANGE

Description Clears CHAP settings.

- This command clears the CHAP username and secret from both initiator and target
- This command clears the CHAP authentication type and sets the host CHAP to NONE

Syntax

```
host.clear_chap host=HOST
```

Arguments

host	Name of an existing host

Example

```
host.clear_chap host=h1
```

Output

```
Host "h1" CHAP settings cleared
```

host.create ADMIN POOL ADMIN CHANGE

Description Create a new host.

Syntax

```
host.create name=NAME
```

Arguments

name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.

Example

```
host.create name=host1
```

Output

```
Host host1 created
```

host.delete ADMIN POOL ADMIN CHANGE

Description Delete host.

Syntax

```
host.delete host=HOST
```

Arguments

host	Name of an existing host

Example

```
host.delete name=host1
```

Output

```
Host host1 deleted
```

host.initiator_query ADMIN POOL ADMIN CHANGE

Description List logged-in initiators which are not assigned to any host in the system.

- Run the command with no parameter to return both FC and ISCSI initiators
- Run the command using the protocol parameter in order to filter out the query results

Syntax

```
host.initiator_query [protocol=TYPE]
```

Arguments

protocol	iSCSI or FC

Query for FC initiators

Example

```
host.initiator_query protocol=FC
```

Output

WWPN	PROTOCOL
21000024ff3cee01	FC
21000024ff3cf0cc	FC

Query for ISCSI initiators

Example

```
host.initiator_query protocol=ISCSI
```

Output

WWPN	PROTOCOL
iqn.1994-05.com.infinidat:iscsi-man03	ISCSI
iqn.1994-05.com.infinidat:iscsi-man02	ISCSI
iqn.1994-05.com.infinidat:iscsi-man04	ISCSI
iqn.1994-05.com.infinidat:iscsi-man01	ISCSI

host.map ADMIN POOL ADMIN

CHANGE

Description Map a volume or a snapshot to a host.

Syntax

```
host.map host=HOST [vol=VOL[,VOL,...]] [snap=SNAP[,SNAP,...]] [lun=LUN]
```

Arguments

lun	SCSI logical unit number (LUN)
snap	Name of an existing volume snapshot (multiple values, separated by commas)
vol	Name of an existing volume (multiple values, separated by commas)
host	Name of an existing host

Example

```
host.map name=host1 vol=vol1
```

Output

```
Volume vol1 mapped to LUN LUN1 in host host1
```

host.map_query ALL ROLES

CHANGE

Description List existing mappings for hosts (clustered hosts included).

Syntax

```
host.map_query [host=HOST] [map_type=HOSTMAPPINGTYPE]
```

Arguments

map_type	CLUSTER or HOST
host	Name of an existing host

Example

```
host.map_query
```

Output

HOST	VOLUME	LUN ID	MAPPING TYPE
h1	vol1	1	HOST
h2	vol2	11	CLUSTER

host.port_query ALL ROLES

CHANGE

Description List ports of existing hosts.

Hosts with no ports are not listed.

Syntax

```
host.port_query [host=HOST] [port=SCSIADDRESS]
```

Arguments

port	A WWN or IQN address. A WWN address consists of a 16-digit hexadecimal number, optionally delimited by colons to 8 groups of two digits each. An IQN address begins with the string "iqn.", optionally followed by a domain registration date, followed by an organizational naming authority, and optionally followed by a colon and a custom string of choosing, e.g "iqn.2001-04.com.example:diskarrays-sn-a8675309".
host	Name of an existing host

An output example for an FC host

Example

```
host.port_query name=host1
```

Output

HOST	TYPE	PORT	LOGGED IN
host1	FC	1000000c9c06f340	yes

An output example of an iSCSI host

Example

```
host.port_query
```

Output

NAME	TYPE	PORT	LOGGED IN
interop006	ISCSI	iqn.1991-05.com.microsoft:interop006	yes
interop008	ISCSI	iqn.1991-05.com.microsoft:interop008	no

host.query ALL ROLES

CHANGE

Description List existing hosts.

Syntax

```
host.query [host=HOST[,HOST,...]] [cluster=CLUSTER[,CLUSTER,...]] [auth_type=METHOD[,METHOD,...]]
```

Arguments

host	Name of an existing host (multiple values, separated by commas)
cluster	Name of an existing cluster (multiple values, separated by commas)
auth_type	Authentication method required of host when connecting through iSCSI: NONE, CHAP or MUTUAL_CHAP (multiple values, separated by commas)

Running the command for with no parameters

Example

```
host.query
```

Output

NAME	CLUSTER	LUNS	FC PORTS	ISCSI IQNS	AUTH TYPE	CREATED AT
iscsi-man01	-	3	0	1	NONE	2016-05-10 10:00:00
iscsi-man02	-	3	0	1	NONE	2016-05-10 11:00:00

Running the command in detailed mode

Example

```
host.query name=iscsi-man01 --detailed
```

Output

Name	iscsi-man01
Cluster	-
LUNs	3
Allocated Capacity	32.98 TB
FC Ports	0
iSCSI IQNs	1
Auth Type	NONE
CHAP Target User	-
CHAP Initiator User	-
Created At	2016-07-25 10:00:00
Updated AT	2016-07-25 10:00:00

host.remove_port

ADMIN POOL ADMIN

CHANGE

Description Remove port from host.

Syntax

```
host.remove_port host=HOST port=DEFINEDSCSIADDRESS[,DEFINEDSCSIADDRESS,...] [type=TYPE]
```

Arguments

port	A WWN or IQN address. A WWN address consists of a 16-digit hexadecimal number, optionally delimited by colons to 8 groups of two digits each. An IQN address begins with the string "iqn.", optionally followed by a domain registration date, followed by an organizational naming authority, and optionally followed by a colon and a custom string of choosing, e.g "iqn.2001-04.com.example:diskarrays-sn-a8675309". (multiple values, separated by commas)
host	Name of an existing host
type	iSCSI or FC

Example

```
host.remove_port name=host1 port=1000000c9c06f340
```

Output

```
FC port 1000000c9c06f340 removed from host host1
```

host.rename

ADMIN POOL ADMIN

CHANGE

Description Rename host.

Syntax

```
host.rename host=HOST new_name=NAME
```

Arguments

new_name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#{}%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
host	Name of an existing host

Example

```
host.rename name=host1 new_name=host2
```

Output

```
Host host1 renamed to host2
```

host.set_auth_type ADMIN POOL ADMIN

CHANGE

Description **Set iSCSI host authentication method.**

- Inbound secret - the key with which the host authenticates with InfiniBox
- Outbound secret - the key with which InfiniBox authenticates with the host

Available authentication types:

- CHAP - the host has to authenticate with InfiniBox
- MUTUAL_CHAP - both host and InfiniBox have to authenticate to each other
- NONE - no authentication is required

Syntax

```
host.set_auth_type host=HOST type=METHOD
```

Arguments

host	Name of an existing host
type	Authentication method required of host when connecting through iSCSI: NONE, CHAP or MUTUAL_CHAP

Setting the authentication method of an iSCSI host to CHAP

Example

```
host.set_auth_type name=h1 type=chap
```

Output

```
Host "h1" iSCSI authentication type set to CHAP, auto-generated inbound secret:
'MVqaQWmB572mB0xzeito'
```

Setting the authentication method of an iSCSI host to MUTUAL_CHAP

Example

```
host.set_auth_type name=h1 type=MUTUAL_CHAP
```

Output

```
Host "h1" iSCSI authentication type set to MUTUAL_CHAP, auto-generated inbound secret:
'cbAl0V2yk6mhLEyV3PAt', auto-generated outbound secret: 'yYqXkGNzxR8CZIjQpjii'
```

Removing an existing authentication requirement

Example

```
host.set_auth_type name=h1 type=NONE
```

Output

```
Host "h1" iSCSI authentication type set to NONE
```

host.set_chap ADMIN POOL ADMIN CHANGE

Description Set iSCSI host CHAP usernames and secrets.

Syntax

```
host.set_chap host=HOST [inbound_user=STR] [inbound_secret=STR] [outbound_user=STR]
[outbound_secret=STR]
```

Arguments

outbound_secret	Any sequence of characters
outbound_user	Any sequence of characters
inbound_secret	Any sequence of characters
inbound_user	Any sequence of characters
host	Name of an existing host

Setting the host authentication without details

Example

```
host.set_chap name=h1
```

Output

```
Host "h1" CHAP settings updated
```

Setting the inbound name and secret

Example

```
host.set_chap name=h1 inbound_user=admin inbound_secret=cbA10V2yk6mhLEyV3PAe
```

Output

```
Host "h1" CHAP settings updated
```

host.unmap ADMIN POOL ADMIN CHANGE

Description Unmap a volume or a snapshot from a host.

Syntax

```
host.unmap host=HOST [vol=VOL[,VOL,...]] [snap=SNAP[,SNAP,...]] [lun=LUN[,LUN,...]]
```

Arguments

lun	SCSI logical unit number (LUN) (multiple values, separated by commas)
snap	Name of an existing volume snapshot (multiple values, separated by commas)
vol	Name of an existing volume (multiple values, separated by commas)
host	Name of an existing host

Example

```
host.unmap name=host1 vol=vol1
```

Output

```
Volume vol1 unmapped from LUN LUN1 in host host1
```

Chapter 11. Cluster

The following cluster commands allow you to create a cluster, to add a host to a cluster and to query the hosts by cluster, to map a volume, snapshot or clone to the host and to query the mapping.

cluster.add_host ADMIN POOL ADMIN

CHANGE

Description Add host to cluster.

Syntax

```
cluster.add_host cluster=CLUSTER host=HOST[,HOST,...]
```

Arguments

host	Name of an existing host (multiple values, separated by commas)
cluster	Name of an existing cluster

Example

```
cluster.add_host name=cluster1 host=host1
```

Output

```
Host host1 added to cluster cluster1
```

cluster.create ADMIN POOL ADMIN

CHANGE

Description Create a new cluster.

Syntax

```
cluster.create name=NAME
```

Arguments

name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.

Example

```
cluster.create name=cluster1
```

Output

```
Cluster cluster1 created
```

cluster.delete ADMIN POOL ADMIN

CHANGE

Description Delete cluster.

Syntax

```
cluster.delete cluster=CLUSTER
```

Arguments

cluster	Name of an existing cluster

Example

```
cluster.delete name=cluster1
```

Output

```
Cluster cluster1 deleted
```

cluster.host_query ALL ROLES

CHANGE

Description List existing clusters with member hosts.

Syntax

```
cluster.host_query [cluster=CLUSTER[,CLUSTER,...]] [host=HOST[,HOST,...]]
```

Arguments

`host`	Name of an existing host (multiple values, separated by commas)
`cluster`	Name of an existing cluster (multiple values, separated by commas)

Example

```
cluster.host_query name=cluster1
```

Output

NAME	HOST
cluster1	host1

cluster.map

ADMIN POOL ADMIN

CHANGE

Description Map a volume or a snapshot to a cluster.

Syntax

```
cluster.map cluster=CLUSTER [vol=VOL[,VOL,...]] [snap=SNAP[,SNAP,...]] [lun=LUN]
```

Arguments

`cluster`	Name of an existing cluster
`lun`	SCSI logical unit number (LUN)
`snap`	Name of an existing volume snapshot (multiple values, separated by commas)
`vol`	Name of an existing volume (multiple values, separated by commas)

Example

```
cluster.map name=cluster1 vol=vol1
```

Output

```
Volume vol1 mapped to LUN LUN1 in cluster cluster1
```

cluster.map_query

ALL ROLES

CHANGE

Description List existing cluster mappings.

Syntax

```
cluster.map_query [cluster=CLUSTER]
```

Arguments

`cluster`	Name of an existing cluster

Example

```
cluster.map_query
```

Output

CLUSTER	VOLUME	LUN
cluster1	vol1	11

cluster.query

ALL ROLES

CHANGE

Description List existing clusters.

This command displays the number of hosts and LUNs for each cluster, along with the cluster creation date and the last time it was modified.

Syntax

```
cluster.query [cluster=CLUSTER[,CLUSTER,...]]
```

Arguments

`cluster`	Name of an existing cluster (multiple values, separated by commas)

Example

```
cluster.query
```

Output

NAME	HOSTS	LUNS	CREATED AT	UPDATED AT
cluster1	2	2	2014-01-01 10:00:00	2014-01-01 10:00:00

cluster.remove_host

ADMIN POOL ADMIN

CHANGE

Description Remove host from cluster.

Syntax

```
cluster.remove_host cluster=CLUSTER host=HOST[,HOST,...]
```

Arguments

`cluster`	Name of an existing cluster
`host`	Name of an existing host (multiple values, separated by commas)

Example

```
cluster.remove_host name=cluster1 host=host1
```

Output

```
Host host1 removed from cluster cluster1
```

cluster.rename

ADMIN POOL ADMIN

CHANGE

Description Rename cluster.

Syntax

```
cluster.rename cluster=CLUSTER new_name=NAME
```

Arguments

`new_name`	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
`cluster`	Name of an existing cluster

Example

```
cluster.rename name=cluster1 new_name=cluster2
```

Output

```
Cluster cluster1 renamed to cluster2
```

cluster.unmap

ADMIN POOL ADMIN

CHANGE

Description Unmap a volume or a snapshot from a cluster.

Syntax

```
cluster.unmap cluster=CLUSTER [vol=VOL[,VOL,...]] [snap=SNAP[,SNAP,...]] [lun=LUN]
```

Arguments

lun	SCSI logical unit number (LUN)
snap	Name of an existing volume snapshot (multiple values, separated by commas)
vol	Name of an existing volume (multiple values, separated by commas)
cluster	Name of an existing cluster

Example

```
cluster.unmap name=cluster1 vol=vol1
```

Output

```
Volume vol1 unmapped from cluster cluster1
```

Chapter 12. Map

The map.query command lists all host and cluster mappings of the InfiniBox volumes and snapshots.

map.query

ALL ROLES

CHANGE

Description **List existing host and cluster mappings.**

You can view results for hosts only by using the host.map_query command.

Syntax

```
map.query
```

Example

```
map.query
```

Output

MAPPED TO	MAPPING TYPE	LUN	VOLUME	SIZE
cluster1	CLUSTER	11	v1	1.00 TB
host1	HOST	1	v2	1.00 TB

Chapter 13. Event

Controls the way InfiniBox events are displayed.

event.codes

ALL ROLES

CHANGE

Description **List all event codes.**

Filter the list by code, level or reporter. State the code to list only the event codes without their details. State the level and select among critical, error, warning and info. State the reporter and select among block, custom, file, management, platform and RMR. Run the command without parameters to get a full list of all events.

Syntax

```
event.codes [code=CODE[,CODE,...]] [level=LEVEL[,LEVEL,...]] [reporter=REPORTER[,REPORTER,...]]
```

Arguments

level	Event severity level (multiple values, separated by commas)
code	Event code (multiple values, separated by commas)
reporter	Event reporter (multiple values, separated by commas)

Listing event codes that contain a specific string. The string can be part of either the code or the description

Example

```
event.codes level=WARNING
```

Output

CODE	LEVEL	REPORTER	DESCRIPTION
ACTIVATION_PAUSED	WARNING	MGMT	System activation paused due to configuration tunable
AUTHENTICATION_FAILURE	WARNING	MGMT	Authentication attempt of user '{username}' from IP:'{client_ip}' failed, please verify the credentials

---

Listing event codes using the grep operator

Example

```
event.codes --grep=snapshot
```

Output

CODE	LEVEL	REPORTER	DESCRIPTION
FILESYSTEM_SNAPSHOT_CREATED	INFO	MGMT	Filesystem snapshot '{FS_Snapshot_Name}' was created for filesystem '{fs_name}'

Listing event codes using the grep operator, for a string that includes a space

Example

```
event.codes --grep='consistency group'
```

Output

CODE	LEVEL	REPORTER	DESCRIPTION
CG_CREATED	INFO	MGMT	Created consistency group '{cg_name}'
CG_DELETED	INFO	MGMT	Deleted consistency group '{cg_name}'

event.create ADMIN CHANGE

Description Create a new custom event.

Set the level and text of the event you create. Then, run the event.query command to see the event ID. Then, use the event ID to run the event.details command.

Syntax

```
event.create level=LEVEL description=STR
```

Arguments

description	Any sequence of characters
level	Event severity level

Example

```
event.create level=INFO description='free text'
```

Output

```
Event created
```

event.details ADMIN POOL ADMIN READ-ONLY CHANGE

Description Display an individual event.

Syntax

```
event.details seq=INT
```

Arguments

seq	An integer number

Example

```
event.details seq=553
```

Output

Affected Entity ID	0
CODE	CUSTOM_INFO_EVENT
Description	GENERAL
Seq	553
Level	INFO
Reporter	CUSTOM
Source Node ID	1
System Version	1.5.0.7
Timestamp	2014-01-01 10:00:00
Username	admin

event.levels

ALL ROLES

CHANGE

Description List all event levels.

Syntax

```
event.levels
```

Example

```
event.levels
```

Output

CRITICAL
ERROR
INFO
WARNING

event.query

ADMIN POOL ADMIN READ-ONLY

CHANGE

Description List system events.

Filter the query by timestamp, ID, code, level or reporter.

Syntax

```
event.query [from=TIMESTAMP] [to=TIMESTAMP] [from_seq=INT] [to_seq=INT] [code=CODE[,CODE,...]]
[exclude=CODE[,CODE,...]] [level=LEVEL[,LEVEL,...]] [reporter=REPORTER[,REPORTER,...]]
[username=USER[,USER,...]]
```

Arguments

from	A timestamp, e.g: '2011-04-13', '2011-04-13 18:30:12'
username	INTERNAL, or name of an existing user (multiple values, separated by commas)
reporter	Event reporter (multiple values, separated by commas)
level	Event severity level (multiple values, separated by commas)
exclude	Event code (multiple values, separated by commas)
code	Event code (multiple values, separated by commas)
to_seq	An integer number
from_seq	An integer number
to	A timestamp, e.g: '2011-04-13', '2011-04-13 18:30:12'

Example

```
event.query from_id=549 to_id=553
```

Output

TIMESTAMP	ID	LEVEL	REPORTER	CODE	USERNAME	DESCRIPTION
2014-05-19 08:40:00	553	INFO	CUSTOM	CUSTOM_INFO_EVENT	admin	GENERAL
2014-05-19 08:30:00	552	INFO	MGMT	CONFIGURATION_PARAMETER_UPDATED	admin	System configuration parameter with category MGMT and key environment.dns_servers changed to 192.168.8.20
2014-05-19 08:20:00	551	INFO	MGMT	NOTIFICATION_TARGET_UPDATED	admin	SYSLOG notification target target0 updated
2014-05-19 08:10:00	550	INFO	MGMT	NOTIFICATION_TARGET_CREATED	admin	SNMP notification target snmpv3 created
2014-05-19 08:00:00	549	INFO	MGMT	USER_LOGIN_SUCCESS	admin	User admin successfully logged in from IP '172.16.4.30'

event.reporters

ALL ROLES

CHANGE

Description List all event reporters.

Syntax

```
event.reporters
```

Example

```
event.reporters
```

Output

REPORTER
BLOCK
CUSTOM
FILE
MGMT
PLATFORM
RMR

event.watch ALL ROLES

CHANGE

Description Monitor system events in real time.

Displays the latest events. The display is refreshed at a rate determined by the Interval parameter. Use the Code and Exclude parameters to determine which events are displayed on the watch. Use Control+C to return to the prompt.

Syntax

```
event.watch [interval=TIMEDELTA] [code=CODE[,CODE,...]] [exclude=CODE[,CODE,...]]
[level=LEVEL[,LEVEL,...]] [reporter=REPORTER[,REPORTER,...]] [username=USER[,USER,...]]
[show_polling=YESNO] [tail_length=TAILLENGTH]
```

Arguments

show_polling	Either yes, or no
username	INTERNAL, or name of an existing user (multiple values, separated by commas)
reporter	Event reporter (multiple values, separated by commas)
level	Event severity level (multiple values, separated by commas)
exclude	Event code (multiple values, separated by commas)
code	Event code (multiple values, separated by commas)
interval	A time delta, e.g: '02:02', '05:11:06'
tail_length	An integer number between 1 and 50 (inclusive)

Example

```
event.watch level=WARNING
```

Output

```
See the documentation of the event.query command.
```

Setting the event watch length

Example

```
event.watch tail_length=10
```

Output

```
See the documentation of the event.query command.
```

Chapter 14. Event Rule

Controls the way events are sent from the InfiniBox to the user.

event.rule.create ADMIN

CHANGE

Description Create a new event notification rule.

Syntax

```
event.rule.create name=NAME [level=LEVEL[,LEVEL,...]] [include=CODE[,CODE,...]]
[exclude=CODE[,CODE,...]] [email=LOCALEMAIL[,LOCALEMAIL,...]] [snmp=SNMPTARGET]
[syslog=SYSLOGTARGET]
```

Arguments

exclude	Event code (multiple values, separated by commas)
include	Event code (multiple values, separated by commas)
level	Event severity level (multiple values, separated by commas)
name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
syslog	Name of an existing syslog notification target
snmp	Name of an existing snmp notification target
email	a valid email address (multiple values, separated by commas)

Example

```
event.rule.create name=rule1include=VOLUME_CREATE,VOLUME_DELETE email=infinibox@infinidat.com
```

Output

```
Event rule rule1 created
```

event.rule.delete ADMIN

CHANGE

Description Delete event notification rule.

Syntax

```
event.rule.delete rule=EVENTRULE
```

Arguments

rule	Name of an existing event rule

Example

```
event.rule.delete name=rule1
```

Output

```
Event rule rule1 deleted
```

event.rule.modify ADMIN

CHANGE

Description Modify existing event notification rule.

Syntax

```
event.rule.modify rule=EVENTRULE [level=LEVEL[,LEVEL,...]] [include=CODE[,CODE,...]]
[exclude=CODE[,CODE,...]] [email=LOCALEMAIL[,LOCALEMAIL,...]] [snmp=SNMPTARGET]
[syslog=SYSLOGTARGET]
```

Arguments

email	a valid email address (multiple values, separated by commas)
exclude	NONE, or event code (multiple values, separated by commas)
include	NONE, or event code (multiple values, separated by commas)
level	NONE, or event severity level (multiple values, separated by commas)
rule	Name of an existing event rule
snmp	Name of an existing snmp notification target
syslog	Name of an existing syslog notification target

Example

```
event.rule.modify name=rule1 include=VOLUME_RESIZE
```

Output

```
Event rule rule1 updated
```

event.rule.query ALL ROLES

CHANGE

Description List existing event rules.

Syntax

```
event.rule.query [rule=EVENTRULE]
```

Arguments

rule	Name of an existing event rule

Example

```
event.rule.query name=rule1
```

Output

Name	rule1
Target	SMTP
Server	customer-smtp
Recipients	recipient@infinidat.com
Level	INFO
Include	VOLUME_CREATE
Exclude	-
Name	rule2
Target	SNMP
Server	snmpv3
Recipients	-
Level	WARNING
Include	VOLUME_DELETE
Exclude	-

event.rule.rename ADMIN

CHANGE

Description Rename existing event notification rule.

Syntax

```
event.rule.rename rule=EVENTRULE new_name=NAME
```

Arguments

new_name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
rule	Name of an existing event rule

Example

```
event.rule.rename name=rule1 new_name=rule2
```

Output

```
Event rule rule1 renamed rule2
```

Chapter 15. Metadata

The meta-data commands allow you to add meta-data to the INfiniBox entities and view meta-data that arrives from entities that were created by the hosts. See examples along the commands documentation.

metadata.clear ADMIN POOL ADMIN

CHANGE

Description Clear all metadata keys from object.

Syntax

```
metadata.clear objtype=OBJTYPE [object=NAME]
```

Arguments

object	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
objtype	vol, vol-snap, fs, fs-snap, pool, host, cluster or system

Example

```
metadata.clear
```

Output

```
All meta-data cleared
```

metadata.get ADMIN POOL ADMIN CHANGE

Description Get metadata from object.

The object type is taken from a predefined list that can be obtained as you type. The object can be obtained from running a query for the specific object type. For example, if you are looking for the metadata of a specific host, query for a host first, and then use host as the object type, and the name of the desired host as an object.

Syntax

```
metadata.get objtype=OBJTYPE [object=NAME] [key=STR]
```

Arguments

object	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
objtype	vol, vol-snap, fs, fs-snap, pool, host, cluster or system
key	Any sequence of characters

Example

```
metadata.get objtype=host object=io004
```

Output

KEY	VALUE
hostname	io004.lab.il.infinidat.com
platform	Linux-3.10.0-123.20.1.el7.x86_64-x86_64-with-centos-7.0.1406-Core
powertools_version	1.13.3
system	Linux

metadata.remove ADMIN POOL ADMIN CHANGE

Description Remove a single metadata key from object.

See the documentation of metadata.get above.

Syntax

```
metadata.remove objtype=OBJTYPE key=NAME [object=STR]
```

Arguments

`object`	Any sequence of characters
`key`	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
`objtype`	vol, vol-snap, fs, fs-snap, pool, host, cluster or system

Example

```
metadata.remove objtype=host object=io004 key=platform
```

Output

```
Meta-data removed
```

metadata.set ADMIN POOL ADMIN

CHANGE

Description Set metadata on object.

You can set one key at a time. setting a value to an already existing key overrides the previous string.

Syntax

```
metadata.set objtype=OBJTYPE key=STR value=STR [object=NAME]
```

Arguments

`objtype`	vol, vol-snap, fs, fs-snap, pool, host, cluster or system
`object`	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
`value`	Any sequence of characters
`key`	Any sequence of characters

Example

```
metadata.set objtype=host object=io004 key=platform value=Linux-3.10.0-123.20.1.el7.x86_64-x86_64-with-centos-7.0.1406-Core
```

Output

```
Meta-data set
```

Chapter 16. System

System commands

system.boot_progress ALL ROLES

CHANGE

Description Display the system boot progress.

Syntax

```
system.boot_progress
```

Example

```
system.boot_progress
```

Output

```
System state: ACTIVE (System is active)
```

system.info ALL ROLES

CHANGE

Description Display general system information.

Syntax

```
system.info
```

Running the command with no operators

Example

```
system.info
```

Output

Name	ibox3000
Serial	3000
Product ID	INFINIBOX
Model Name	F6100
Version	2.2.0.9
Uptime	185:38:48 hours
Operational State	ACTIVE
WWNN	57:42:b0:f0:00:0b:bb:00
SSD cache drives	0
Total Physical Capacity	1383.28 TB
Free Physical Capacity	210.91 TB
Pools Physical Capacity	409.97 TB
Pools Allocated Physical Capacity	17.84 TB
Total Virtual Capacity	1383.30 TB
Free Virtual Capacity	418.78 TB
Pools Virtual Capacity	18.93 TB
Pools Allocated Virtual Capacity	17.84 TB
Compression Savings	2.55 : 1
Pools	131
Volumes	15452 (31005 snapshots)
Filesystems	9863 (20110 snapshots)
Consistency Groups	2 (3 snap-groups)
Replicas	5
Exports	1052
Hosts	33
Inactive Drives	0
Rebuild 1 in progress	no
Rebuild 2 in progress	no

Running the command using the grep operator

Example

```
system.info --grep=replica
```

Output

Replicas	**2**

system.show_metrics ALL ROLES

CHANGE

Description System overall traffic

This command displays the overall system traffic, broken down to NAS and SAN. For detailed traffic metrics, look for metrics.* commands.

Syntax

```
system.show_metrics [interval=Metrics Interval] [count=POSITIVE]
```

Arguments

interval	Metrics refresh interval. Max value is 60
count	A positive integer number, greater than zero

Example

```
system.show_metrics
```

Output

TYPE	OPERATIONS	THROUGHPUT	LATENCY
NAS	35,438	523 MB/Sec	1.7 ms
SAN	15,023	1,233 MB/Sec	1.2 ms

system.shutdown ADMIN

CHANGE

Description Initiate system shutdown process.

Syntax

```
system.shutdown
```

Example

```
system.shutdown
```

Output

```
Warning: You are about to shutdown the system. This will stop all IO services to hosted volumes. To avoid potential loss of data, please make sure that no application is accessing the system's volumes. Are you sure you wish to continue? y/n
```

system.upgrade ADMIN

CHANGE

Description Upload an upgrade package.

Syntax

```
system.upgrade filename=STR
```

Arguments

filename	Any sequence of characters

Example

```
system.upgrade filename=upgrade-file-name
```

Output

Chapter 17. System Service

system.service.query ADMIN

CHANGE

Description report the state of InfiniBox services.

Syntax

```
system.service.query [services=SERVICE[,SERVICE,...]]
```

Arguments

services	Name of an existing service (multiple values, separated by commas)

Example

```
system.service.query
```

Output

SERVICE	STATE
core	ACTIVE
iscsi	ACTIVE
management	ACTIVE
nas	ACTIVE
replication	ACTIVE

Chapter 18. Replica

replica.change_role ADMIN POOL ADMIN CHANGE

Description **Change the replica role on the local system only.**

Changes the role of the local replica from source to target, or from target to source. When using this command in a failover scenario you have to run this command twice, once on the source and once on the target.

Syntax

```
replica.change_role local_dataset=REPLICATEDENTITY [dataset_type=CGORDATASET]
```

Arguments

local_dataset	Name of an existing dataset or consistency group with replication
dataset_type	CG, VOLUME or FILESYSTEM

Example

```
replica.change_role dataset=vol1
```

Output

```
Replica rep1 role is now TARGET, and its staging area was discarded
```

replica.create ADMIN POOL ADMIN CHANGE

Description Create a new replica.

This command creates a new replica.

- Set the replica type - either SYNC or ASYNC
- Set the remote system
- Set the local entity (volume, consistency group, or filesystem)
- Set the local entity type
- Set an existing remote entity
    - Otherwise, set the remote pool
        - Optionally set the name of the remote entity that will be created
        - Optionally set a suffix for the target consistency group members
- For ASYNC replica - set the interval and RPO in seconds, both values have to be between 4 seconds and 24 hours.

A note on name uniqueness:

- Two replicated entities (of any entity types) should not have the same name.

Syntax

```
replica.create system=SYSTEM source=NONREPLICATEDENTITY remote_pool=REMOTEENTITY
replication_type=REPLICATYPE [interval=TIMEDELTA] [rpo=TIMEDELTA] [dataset_type=CGORDATASET]
[new_target_name=NAME] [new_datasets_suffix=NAME]
```

Arguments

`source`	Name of an existing dataset or consistency group without replication
`system`	Name of an existing replication link
`new_datasets_suffix`	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@()[]$=!-#{}%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
`new_target_name`	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@()[]$=!-#{}%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
`dataset_type`	CG, VOLUME or FILESYSTEM
`rpo`	A time delta, e.g: '02:02', '05:11:06'
`interval`	A time delta, e.g: '02:02', '05:11:06'
`replication_type`	ASYNC or SYNC
`remote_pool`	Name of a remote entity (auto-completion unavailable)

Create an ASYNC replica for a volume.

Example

```
replica.create system=localsystem source=vol-1 interval=60 rpo=120
```

Output

```
Replica for volume vol-1 created
```

Create a replica with no interval. When a replica is created with no Interval, a sync job can be created only manually.

Example

```
replica.create system=localsystem source=vol-2 interval=NONE
```

Output

```
Replica for volume vol-2 created
```

Create a replica with no RPO. When a replica is created with no RPO, the InfiniBox system does not manage the RPO states, nor create events for the transition from RPO OK to RPO lagging and back.

Example

```
replica.create system=localsystem source=vol-3 rpo=NONE
```

Output

```
Replica for volume vol-3 created
```

Create a replica for a consistency group. The interval and RPO are set to 1 minute each.

Example

```
replica.create system=localsystem source=cg-1 interval=60 rpo=60
```

Output

```
Replica for consistency group cg-1 created
```

Create a synchronous replica

Example

```
replica.create system=ibox628 remote_pool=p1 source=vol1000 replication_type=sync
```

Output

```
Replica for volume vol1000 created
```

replica.delete ADMIN POOL ADMIN CHANGE

Description **Delete a replica.**

Delete a replica by indicating the source dataset. The replica is deleted on both source and target.

- For ASYNC replica: indicate whether to keep the staging area.
- Use the FORCE flag to delete only the local replica (for example, in a case of link down)

Syntax

```
replica.delete local_dataset=REPLICATEDENTITY [dataset_type=CGORDATASET] [retain_staging_area=YESNO] [force=YESNO]
```

Arguments

`retain_staging_area`	Either yes, or no
`dataset_type`	CG, VOLUME or FILESYSTEM
`local_dataset`	Name of an existing dataset or consistency group with replication
`force`	Either yes, or no

Example

```
replica.delete dataset=vol1 retain_staging_area=yes
```

Output

```
Replica replica1 deleted
```

replica.query ALL ROLES CHANGE

Description **List existing replicas.**

Use the local_dataset parameter in order to filter the output by the replicated dataset(s). The query can be filtered by:

- replication_type
- state

Syntax

```
replica.query [system=SYSTEM] [local_dataset=REPLICATEDENTITY[,REPLICATEDENTITY,...]] [role=ROLE]
```

Arguments

role	SOURCE or TARGET
local_dataset	Name of an existing dataset or consistency group with replication (multiple values, separated by commas)
system	Name of an existing replication link

Querying replicas - synchronous example

Example

```
replica.query replication_type=sync
```

Output

LOCAL DATASET	TYPE	ROLE	REMOTE SYSTEM	REMOTE DATASET	STATE	SYNC STATE	LATENCY
vol_90b000	VOLUME	SOURCE	ibox3000	vol_dbad75	ACTIVE	SYNCHRONIZED	0.0 ms
vol_c9be33	VOLUME	SOURCE	ibox3000	vol_45f609	ACTIVE	INITIALIZING	0.0 ms

---

Qurying replicas - asynchronous example

Example

```
replica.query replication_type=async
```

Output

LOCAL DATASET	TYPE	ROLE	REMOTE SYSTEM	REMOTE DATASET	STATE	JOB STATE	PROGRESS	STATE	RPO STATE	RESTORE POINT
cg-1	CG	SOURCE	ibox3000	cg-1-remote	ACTIVE	DONE	100%	ACTIVE	OK	2015-08-01 10:00:00

Querying for replicated datasets with ongoing sync jobs.

Example

```
replica.query --grep=replicating --
columns=local_dataset,state,progress,throughput,rpo_state,interval,rpo_value
```

Output

LOCAL DATASET	STATE	PROGRESS	THROUGHPUT	RPO STATE	INTERVAL	RPO VALUE
io004_replica_to_ibox1106	REPLICATING	90%	249 MB/sec	OK	0:00:05	30 sec
io004_replica_to_ibox1177	REPLICATING	4%	233 MB/sec	OK	0:00:05	30 sec
io004_replica_to_ibox3008	REPLICATING	10%	240 MB/sec	OK	0:00:05	30 sec

---

Querying for replicas, displaying a detailed output

Example

```
replica.query --detailed
```

Output

Local Dataset	cg1
Local Pool Name	p1
Role	SOURCE
Remote System	box-avi
Remote Dataset	cg1-box-avi
Remote Pool Name	p1-box-avi
State	ACTIVE
State Description	-
Job State	DONE
Progress	100%
Throughput	-
RPO State	OK
RPO Value	0:05:00
Sync Interval	0:01:00
Created At	2016-09-12 10:00:00
Updated At	2016-09-12 11:00:00
Last Synchronized	2016-09-12 11:00:02
Restore Point	2016-09-12 11:00:02
Staging Area Size	0
Pairs	

Local Dataset	Remote Dataset	Initializing	Restore Point
v-1	v-1-target	No	2016-09-12 11:20:00
v-2	v-2-target	No	2016-09-12 11:20:00

Querying replicas - asynchronous example

Example

```
replica.query
```

Output

Malformed JSON

Filter the query by the state of the replica

Example

```
replica.query state=SUSPENDED  --columns=local_dataset,state
```

Output

LOCAL DATASET	STATE
vol1	SUSPENDED
LOCAL DATASET	STATE
vol2	SUSPENDED

replica.resume ADMIN POOL ADMIN CHANGE

Description Resumes the replica.

This command resumes the replica from a SUSPENDED or AUTO-SUSPENDED state. The command has to be run on the source only.

Syntax

```
replica.resume [local_dataset=REPLICATEDENTITY] [dataset_type=CGORDATASET] [resume_all=YESNO]
```

Arguments

local_dataset	Name of an existing dataset or consistency group with replication
resume_all	Either yes, or no
dataset_type	CG, VOLUME or FILESYSTEM

Run the command, specifying the local dataset. Then, enter the credentials to the remote system.

Example

```
replica.resume dataset=vol1
Username for box-avi: admin
Password:
```

Output

```
Replica replica1 resumed
```

replica.set_interval ADMIN POOL ADMIN CHANGE

Description **Set the replica sync interval.**

The interval cannot be greater than the RPO.

- This command is available for ASYNC replica only
- This command can be run only from the source

Syntax

```
replica.set_interval local_dataset=REPLICATEDENTITY interval=TIMEDELTA [dataset_type=CGORDATASET]
```

Arguments

dataset_type	CG, VOLUME or FILESYSTEM
interval	A time delta, e.g: '02:02', '05:11:06'
local_dataset	Name of an existing dataset or consistency group with replication

Example

```
replica.set_interval dataset=vol1 interval=60
```

Output

```
Replica for volume vol1 set to synchronize every 60 seconds
```

replica.set_rpo ADMIN POOL ADMIN CHANGE

Description **Set the replica recovery point objective (RPO).**

The RPO has to be greater than the interval.

- This command is available for ASYNC replica only
- This command can be run only from the source

It is recommended to set the RPO to be at least twice as large as the Interval.

Syntax

```
replica.set_rpo local_dataset=REPLICATEDENTITY rpo=TIMEDELTA [dataset_type=CGORDATASET]
```

Arguments

rpo	A time delta, e.g: '02:02', '05:11:06'
local_dataset	Name of an existing dataset or consistency group with replication
dataset_type	CG, VOLUME or FILESYSTEM

Example

```
replica.set_rpo local_dataset=cg-1 rpo=60
```

Output

```
Replica for consistency group cg-1 recovery point objective set to 60 seconds
```

replica.suspend ADMIN POOL ADMIN CHANGE

Description Suspend the replica.

Syntax

```
replica.suspend local_dataset=REPLICATEDENTITY [dataset_type=CGORDATASET]
```

Arguments

local_dataset	Name of an existing dataset or consistency group with replication
dataset_type	CG, VOLUME or FILESYSTEM

Example

```
replica.suspend dataset=vol1
```

Output

```
Replica replica1 suspended
```

replica.switch_role ADMIN POOL ADMIN CHANGE

Description Switch the roles of the replica

Switching the roles of replicated datasets changes the source to target and the target to source.

- When the source switches to target, it becomes write-protected.
- When the target switches to source, it becomes write-enabled (in order to accept host I/O)

The command is available for SYNC replica only. The command has to be run on the source. The replica has to be Synchronized.

Syntax

```
replica.switch_role local_dataset=REPLICATEDENTITY [dataset_type=CGORDATASET]
```

Arguments

dataset_type	CG, VOLUME or FILESYSTEM
local_dataset	Name of an existing dataset or consistency group with replication

Example

```
replica.switch_role local_dataset=vol1001
```

Output

```
Roles of replica vol1001 have changed successfully
```

replica.sync_now ADMIN POOL ADMIN CHANGE

Description Initiate a sync job.

Initiate a sync job that runs immediately. This command assumes that a replica is defined correctly and has already past its Initialization phase. The command runs only on the source dataset.

Syntax

```
replica.sync_now local_dataset=REPLICATEDENTITY [dataset_type=CGORDATASET]
```

Arguments

local_dataset	Name of an existing dataset or consistency group with replication
dataset_type	CG, VOLUME or FILESYSTEM

Initiating a sync for a volume

Example

```
replica.sync_now local_dataset=vol-2
```

Output

```
Replica for volume vol-2 sync started
```

Initiating a sync for a consistency group

Example

```
replica.sync_now local_dataset=cg-1
```

Output

```
Replica for consistency group cg-1 sync started
```

Chapter 19. Replica Advanced

replica.advanced.create ADMIN POOL ADMIN CHANGE

Description **Create a new replica using a target dataset that already exists on the remote system.**

This command uses a dataset or a snapshot as a baseline. Using a source snapshot and a target snasphot as a baseline, both snapshots need to be retained from the previous replica.

Syntax

```
replica.advanced.create system=SYSTEM source=NONREPLICATEDDATASET target=REMOTEENTITY
replication_type=REPLICATYPE [interval=TIMEDELTA] [rpo=TIMEDELTA] [dataset_type=CGORDATASET]
[source_snapshot=SNAP] [target_snapshot=REMOTEENTITY]
```

Arguments

target	Name of a remote entity (auto-completion unavailable)
source	Name of an existing dataset without replication
system	Name of an existing replication link
target_snapshot	Name of a remote entity (auto-completion unavailable)
source_snapshot	Name of an existing snapshot
dataset_type	CG, VOLUME or FILESYSTEM
rpo	A time delta, e.g: '02:02', '05:11:06'
interval	A time delta, e.g: '02:02', '05:11:06'
replication_type	ASYNC or SYNC

Replicating a volume using a source volume and a target volume.

Example

```
replica.advanced.create system=ibox3000 source=vol-3 target=vol-3-target interval=60 rpo=60
```

Output

```
Replica from volume vol-3 to volume vol-3-target created
```

Replicating a volume using a source snapshot and a target snapshot.

Example

```
replica.advanced.create system=ibox3000 source=vol-3 source_snapshot=vol-3-s-1 target=vol-3-target
target_snapshot=vol-3-target-s-1 interval=60 rpo=60
```

Output

```
Replica for volume vol-3 created
```

replica.advanced.create_cg ADMIN POOL ADMIN CHANGE

Description **Create a new replica to target consistency group that already exists on remote system.**

This command uses a consistency group or a snapshot group as a baseline. Using a source snapshot group and a target snasphot group as a baseline, both snapshot groups need to be retained from the previous replica.

Syntax

```
replica.advanced.create_cg system=SYSTEM source=NONREPLICATEDCG interval=TIMEDELTA rpo=TIMEDELTA
target=REMOTEENTITY [cg_pairs=LOCAL:REMOTE[,LOCAL:REMOTE,...]]
```

Arguments

system	Name of an existing replication link
cg_pairs	Pairs of local and remote dataset names, each delimited by a colon, e.g: 'vol01:rep-vol01,vol02:rep-vol02' (multiple values, separated by commas)
target	Name of a remote entity (auto-completion unavailable)
rpo	A time delta, e.g: '02:02', '05:11:06'
interval	A time delta, e.g: '02:02', '05:11:06'
source	Name of an existing consistency group without replication

---

Replicating a consistency group using a source consistency group and a target consistency group.

Example

```
replica.advanced.create_cg system=ibox3000 source=cg-2 target=cg-2-target interval=45 rpo=45
```

Output

```
Replica for consistency group cg-2 created using automatic configuration
```

Replicating a consistency group using source and target consistency groups and pairs of source and target consistency group members.

Example

```
replica.advanced.create_cg system=ibox3000 source=cg-2 target=cg-2-target cg_pairs=vol-11:remote-
vol-11,vol-12:remove-vol-12 interval=45 rpo=45
```

Output

```
Replica for consistency group cg-2 created using automatic configuration
```

Chapter 20. Link

link.attach ADMIN CHANGE

Description Attach network space to link.

Syntax

```
link.attach remote_system=SYSTEM network_space=REPLICATIONNETWORKSPACE
```

Arguments

remote_system	Name of an existing replication link
network_space	Name of an existing replication network space

Example

```
link.attach system=ibox1089 network_space=replication_link
```

Output

```
Replication link "ibox1089" attached
```

link.authenticate ADMIN POOL ADMIN CHANGE

Description Authenticate a link to a remote system.

This command authenticates the link between the current InfiniBox system to a remote system. The command receives the name of the remote system as an input. If user credentials to the remote system are not provided as input, you will be asked to type them.

Syntax

```
link.authenticate remote_system=SYSTEM [username=STR] [password=STR]
```

Arguments

password	Any sequence of characters
username	Any sequence of characters
remote_system	Name of an existing replication link

Example

```
link.authenticate system=ibox3010
```

Output

```
Authenticated on ibox3010 as user user1 with role ADMIN
```

link.change_remote_address ADMIN CHANGE

Description Change remote address.

Syntax

```
link.change_remote_address remote_system=SYSTEM remote_address=HOSTNAME
```

Arguments

remote_address	A valid hostname or IPv4 address
remote_system	Name of an existing replication link

Example

```
link.change_remote_address system=ibox1089 remote_address=172.16.32.211
```

Output

```
The remote address of system "ibox1089" has changed to 172.16.32.211
```

link.create ADMIN CHANGE

Description Create a new replication link.

Syntax

```
link.create remote_address=HOSTNAME network_space=REPLICATIONNETWORKSPACE
```

Arguments

network_space	Name of an existing replication network space
remote_address	A valid hostname or IPv4 address

Example

```
link.create remote_address=ibox3009 network_space=REPLICATION
```

Output

```
Link ibox3009 created
```

link.delete ADMIN

CHANGE

Description Delete replication link.

Syntax

```
link.delete remote_system=SYSTEM [force=YESNO]
```

Arguments

remote_system	Name of an existing replication link
force	Either yes, or no

Example

```
link.delete system=ibox3009
```

Output

```
Link ibox3009 deleted
```

link.detach ADMIN

CHANGE

Description Detach network space from link.

Syntax

```
link.detach remote_system=SYSTEM
```

Arguments

remote_system	Name of an existing replication link

Example

```
link.detach remote_system=ibox1089
```

Output

```
Link is detached from the remote system ibox1089
```

link.query ALL ROLES

CHANGE

Description List existing replication links.

The output can be filtered by a remote system, a network space, or both

Syntax

```
link.query [remote_system=SYSTEM[,SYSTEM,...]] [net_space=NETWORKSPACE]
```

Arguments

net_space	Name of an existing network space
remote_system	Name of an existing replication link (multiple values, separated by commas)

Example

```
link.query net_space=REPLICATION
```

Output

REMOTE SYSTEM	LINK STATE	LAST CONNECTION	LOCAL NETWORK SPACE	REMOTE SYSTEM ADDRESS
ibox3009	UP	2015-03-01 10:00:00	replication_space	172.16.62.36

link.refresh ADMIN

CHANGE

Description Refresh link.

Syntax

```
link.refresh remote_system=SYSTEM
```

Arguments

remote_system	Name of an existing replication link

The link is identified by the service it carries.

Example

```
link.refresh system=ibox1089
```

Output

```
Replication link to system ibox1089 is refreshed
```

Chapter 21. Node

List system nodes.

node.query ALL ROLES

CHANGE

Description List system nodes.

Syntax

```
node.query
```

Example

```
node.query
```

Output

NAME	STATE	CORE STATE	CORE ROLE	MGMT STATE	MGMT ROLE
node-1	ACTIVE	ACTIVE	MASTER	ACTIVE	MASTER

Chapter 22. Node Drive

node.drive.query ALL ROLES

CHANGE

Description List node drives.

Syntax

```
node.drive.query [drive=NODEDRIVE] [node=NODE] [type=Drive Node Type]
```

Arguments

type	Storage device type (DISK or SSD)
node	Node name
drive	Node and drive index, e.g: N1D8

Example

```
node.drive.query
```

Output

NAME	TYPE	STATE	STATE DESCRIPTION
N1D01	Disk	OK	-
N1D02	Disk	OK	-
...			
N1D09	SSD	MISSING	Drive is missing
...			

Chapter 23. Drive

Drive commands manage the process of replacing a disk drive on the InfiniBox.

drive.query ALL ROLES

CHANGE

Description **List all data-drives.**

Lists the InfiniBox drives. Run without parameters to list all of the drives. Use the drive parameter to filter the output.

Syntax

```
drive.query [drive=DRIVE[,DRIVE,...]] [enclosure=ENCLOSURE[,ENCLOSURE,...]]
```

Arguments

enclosure	Enclosure name (multiple values, separated by commas)
drive	Enclosure and drive index, e.g: E2D17 (multiple values, separated by commas)

Example

```
drive.query
```

Output

NAME	STATE	CAPACITY	ACTIVE SINCE
E1D01	ACTIVE	1 TB	2014-01-01 10:00:00
E1D01	ACTIVE	1 TB	2014-01-01 10:00:00

Chapter 24. Enclosure

List all data-drive enclosures.

enclosure.query ALL ROLES

CHANGE

Description **List all data-drive enclosures.**

This command lists the InfiniBox Enclosures along with their state and number of drives. User drive.query in order to investigate the state of each of the drives.

Syntax

```
enclosure.query
```

Example

```
enclosure.query
```

Output

INDEX	STATE	DRIVES
1	OK	60
2	OK	60
3	OK	60
4	OK	60
5	OK	60
6	OK	60
7	OK	60
8	OK	60

Chapter 25. User

User commands allow you to create a user, assign him as a pool admin (and change his role to allow that), map him to LDAP and query all of the users on an InfiniBox.

user.add_pool ADMIN

CHANGE

Description **Grant pool provisioning privileges to a pool administrator.**

This command operates on pool administrators only.

Syntax

```
user.add_pool user=POOLADMINUSER pool=POOL
```

Arguments

user	Name of an existing local user with PoolAdmin role
pool	Name of an existing pool

Example

```
user.add_pool name=user1 pool=pool1
```

Output

```
Granted provisioning privileges on pool pool1 to user user1
```

user.change_email ADMIN

CHANGE

Description **Change local user email.**

Syntax

```
user.change_email user=LOCALUSER email=LOCALEMAIL
```

Arguments

user	Name of an existing local user
email	a valid email address

Example

```
user.change_email name=user1 email=user1@infinidat.com
```

Output

```
User user1 email changed from email1 to email2
```

user.change_password ALL ROLES

CHANGE

Description Change your password.

Run the command, enter a new password and re-enter when prompted to do so.

Syntax

```
user.change_password [user=LOCALUSER] [password_env_variable=ENVVAR]
```

Arguments

password_env_variable	Name of an existing environment variable
user	Name of an existing local user

Example

```
user.change_password
```

Output

```
Local user user1 password changed
```

user.change_role ADMIN CHANGE

Description Change local user role.

Available roles: Admin, PoolAdmin, ReadOnly. The availability of these roles depends on the role of the user that runs this command.

Syntax

```
user.change_role user=LOCALUSER role=ROLE
```

Arguments

user	Name of an existing local user
role	User role

Example

```
user.change_role name=user1 role=pool-admin-1
```

Output

```
User user1 role changed from PoolAdmin to ReadOnly
```

user.create ADMIN CHANGE

Description Create a new local user.

Syntax

```
user.create name=NAME email=LOCALEMAIL role=ROLE [password=STR]
```

Arguments

password	Any sequence of characters
role	User role
email	a valid email address
name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@()[]$=!-#{}%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.

Example

```
user.create name=user1 password=****** email=user1@infinidat.com role=pool-admin-1
```

Output

```
User user1 created
```

user.delete ADMIN

CHANGE

Description Delete local user.

Syntax

```
user.delete user=LOCALUSER
```

Arguments

`user`	Name of an existing local user

Example

```
user.delete name=user1
```

Output

```
User user1 deleted
```

user.disable ADMIN

CHANGE

Description Disables a local user.

This command prevents the user from logging into the system. The user is not deleted and can be re-enabled.

Syntax

```
user.disable user=ENABLEDLOCALUSER
```

Arguments

`user`	Name of an existing enabled local user

Example

```
user.disable name=pool-admin-1
```

Output

```
Local user "pool-admin-1" disabled
```

user.enable ADMIN

CHANGE

Description Enables a local user.

Syntax

```
user.enable user=DISABLEDLOCALUSER
```

Arguments

`user`	Name of an existing disabled local user

Example

```
user.enable name=pool-admin-1
```

Output

```
Local user "pool-admin-1" enabled
```

user.query ALL ROLES

CHANGE

Description List existing local users.

The users are displayed along with their roles and the pools they have access to.

Syntax

```
user.query [user=LOCALUSER[,LOCALUSER,...]] [role=ROLE]
```

Arguments

user	Name of an existing local user (multiple values, separated by commas)
role	User role

Example

```
user.query
```

Output

NAME	ROLE	POOLS
user1	Admin	all pools
user2	PoolAdmin	pool1
user3	PoolAdmin	-

user.remove_pool ADMIN

CHANGE

Description Revoke pool provisioning privileges from a pool administrator.

Syntax

```
user.remove_pool user=POOLADMINUSER pool=POOL
```

Arguments

pool	Name of an existing pool
user	Name of an existing local user with PoolAdmin role

Example

```
user.remove_pool name=user1 pool=pool1
```

Output

```
Revoked provisioning privileges on pool pool1 from user user1
```

user.rename ADMIN

CHANGE

Description Change local user username.

Syntax

```
user.rename user=LOCALUSER new_name=STR
```

Arguments

new_name	Any sequence of characters
user	Name of an existing local user

Example

```
user.rename name=user1 new_name=user2
```

Output

```
User user1 renamed to user2
```

Chapter 26. User Group

user.group.add_pool ADMIN

CHANGE

Description Grant pool provisioning privileges to an LDAP group of pool administrators.

Syntax

```
user.group.add_pool group=POOLADMINGROUP pool=POOL
```

Arguments

`pool`	Name of an existing pool
`group`	Name of an existing ldap group with PoolAdmin role

Example

```
user.group.add_pool name=user-group-1 pool=pool1
```

Output

```
Granted provisioning privileges on pool pool1 to ldap group usergroup-1
```

user.group.change_role ADMIN

CHANGE

Description **Change LDAP or Active Directory group role.**

Change the InfiniBox user role with which the members of an LDAP group access the InfiniBox.

Syntax

```
user.group.change_role group=LDAPGROUP role=ROLE
```

Arguments

`role`	User role
`group`	Name of an existing ldap group

Example

```
user.group.change_role name=group1 role=PoolAdmin
```

Output

```
LDAP group group1 role changed from ReadOnly to PoolAdmin
```

user.group.create ADMIN

CHANGE

Description **Map an LDAP group to an InfiniBox user role.**

Allow members of an LDAP group to access the InfiniBox with a specified user role. The available user roles are: Admin, PoolAdmin and ReadOnly.

Syntax

```
user.group.create name=NAME role=ROLE ldap=USERSREPOSITORY dn=LDAPDN
```

Arguments

`ldap`	Name of an existing ldap server
`role`	User role
`name`	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
`dn`	LDAP distinguished name

Example

```
user.group.create name=ldap1 role=Admin ldap=infinidat.com
dn=CN=Administrators,CN=Builtin,DC=infinidat,DC=com
```

Output

```
LDAP group ldap1 created
```

user.group.delete ADMIN

CHANGE

Description **Unmap an LDAP group from a user role.**

Syntax

```
user.group.delete group=LDAPGROUP
```

Arguments

group	Name of an existing ldap group

Example

```
user.group.delete name=ldap1
```

Output

```
LDAP group ldap1 deleted
```

user.group.query ALL ROLES

CHANGE

Description List LDAP groups that are mapped to user roles.

Syntax

```
user.group.query [group=LDAPGROUP]
```

Arguments

group	Name of an existing ldap group

Example

```
user.group.query
```

Output

NAME	ROLE	LDAP	DN	POOLS
admin_group	Admin	inf	CN=Users,CN=Builtin,DC=corp,DC=mgmttest	
read_only	ReadOnly	read	CN=dev,OU=local,DC=infinidat.com	-

user.group.remove_pool ADMIN

CHANGE

Description Revoke pool provisioning privileges from an LDAP group of pool administrators.

Syntax

```
user.group.remove_pool group=POOLADMINGROUP pool=POOL
```

Arguments

group	Name of an existing ldap group with PoolAdmin role
pool	Name of an existing pool

Example

```
user.group.remove_pool name=user-group-1 pool=pool1
```

Output

```
Revoked provisioning privileges on pool pool1 from ldap group usergroup-1
```

user.group.rename ADMIN

CHANGE

Description Rename a user group.

Syntax

```
user.group.rename group=LDAPGROUP new_name=STR
```

Arguments

new_name	Any sequence of characters
group	Name of an existing ldap group

Example

```
user.group.rename name=user-group-1 new_name=user-group-2
```

Output

```
User group user-group-1 renamed to user-group-2
```

Chapter 27. Config System

System configuration commands set the InfiniBox DNS, Management IP and name.

config.system.query ALL ROLES

CHANGE

Description Display general system configuration.

Syntax

```
config.system.query
```

Example

```
config.system.query
```

Output

Name	ibox505
Management IP Address	172.16.65.75 / 255.255.224.0
Default Gateway	172.16.95.254
SSL Redirection	enabled
Active Session Timeout	1440 Minutes
Idle Session Timeout	60 Minutes
Node 1 IP Address	172.16.67.35
DNS Servers	192.168.8.20
NTP Servers	ntp01.infinidat.com
Message Of The Day	Infinidat’s enterprise storage solutions offer the performance, flexibility and reliability that are necessary to support today’s data intensive enterprises.

config.system.set_default_pool_compression ADMIN

CHANGE

Description Enables / disables default pool compression.

- This command sets the system default to allow pools to use compression for all of their datasets
- This system-level setting can be overridden per pool, using the pool.set_compression command
- This system-level setting can be overridden per dataset, using the following commands:
  - vol.set_compression
  - fs.set_compression

Syntax

```
config.system.set_default_pool_compression compression=YESNO
```

Arguments

compression	Either yes, or no

Example

```
config.system.set_default_pool_compression compression=yes
```

Output

```
Default pool compression set to "enabled"
```

config.system.set_default_pool_ssd_cache ADMIN

CHANGE

Description Enables / disables default pool SSD cache

- This command sets the system default to allow pools to use SSD cache for all of their datasets
- This system-level setting can be overridden per pool, using the pool.cache command
- This system-level setting can be overridden per dataset, using the following commands:
  - vol.cache
  - vol.snap.cache
  - fs.cache
  - fs.snap.cache

Syntax

```
config.system.set_default_pool_ssd_cache ssd_cache=YESNO
```

Arguments

ssd_cache	Either yes, or no

Example

```
config.system.set_default_pool_ssd_cache ssd_cache=yes
```

Output

```
Default pool ssd cache set to "enabled"
```

config.system.set_dns ADMIN

CHANGE

Description Set DNS server IPs.

Syntax

```
config.system.set_dns primary=IPADDRESS [secondary=IPADDRESS]
```

Arguments

primary	A valid IPv4 address
secondary	A valid IPv4 address

Example

```
config.system.set_dns primary=192.168.8.21
```

Output

```
DNS server set to 192.168.8.21
```

config.system.set_motd ADMIN

CHANGE

Description Set the message of the day

- The message is visible via config.system.query.

Syntax

```
config.system.set_motd [clear=YESNO]
```

Arguments

clear	Either yes, or no

- Run the command.
- When prompted, enter the message text. Then, type Ctrl+D to save the message.

Example

```
config.system.set_motd
```

Output

```
Message of the day set to "sequence of characters"
```

config.system.set_name ADMIN CHANGE

Description Set system name.

Syntax

```
config.system.set_name name=NAME
```

Arguments

name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.

Example

```
config.system.set_name name=box-cl002
```

Output

```
System renamed from box-cl001 to box-cl002
```

config.system.set_session_timeout ADMIN CHANGE

Description Set system timeouts.

InfiniBox session timeout values are configurable as follows:

- Idle session timeout - sets the session to expire in case of user inactivity
  - The user's credentials are kept for the timeout period. Within this period, when opening the GUI in the browser, there is no need to type the user and password.
  - By default, the session terminates after 1 hour of inactivity
- Active session timeout - sets the session to disconnect the user
  - By default, the session expires after 24 hours

Syntax

```
config.system.set_session_timeout [active_session=SESSIONTIMEOUT] [idle_session=SESSIONTIMEOUT]
```

Arguments

idle_session	Session timeout. In minutes
active_session	Session timeout. In minutes

Example

```
config.system.set_session_timeout active_session=2880 idle_session=1440
```

Output

```
Session expiry set to "2880 minutes", Session idle timeout set to "1440 minutes"
```

Chapter 28. Config System Time

config.system.time.set_ntp ADMIN CHANGE

Description Set NTP servers.

Syntax

```
config.system.time.set_ntp primary=HOSTNAME [secondary=HOSTNAME]
```

Arguments

secondary	A valid hostname or IPv4 address
primary	A valid hostname or IPv4 address

Example

```
config.system.time.set_ntp primary=ntp1
```

Output

```
NTP server set to ntp1
```

Chapter 29. Config FC Port

config.fc.port.disable ADMIN

CHANGE

Description Disable FC port.

Syntax

```
config.fc.port.disable port=PORT
```

Arguments

port	Node and FC port index, e.g: N2FC5, or WWPN (a 16-digit hexadecimal number, optionally divided by colons to 8 groups of two digits each)

Example

```
config.fc.port.disable name=N1FC1
```

Output

```
FC port N1FC1 disabled
```

config.fc.port.enable ADMIN

CHANGE

Description Enable FC port.

Syntax

```
config.fc.port.enable port=PORT
```

Arguments

port	Node and FC port index, e.g: N2FC5, or WWPN (a 16-digit hexadecimal number, optionally divided by colons to 8 groups of two digits each)

Example

```
config.fc.port.enable name=N1FC1
```

Output

```
FC port N1FC1 enabled
```

config.fc.port.query ALL ROLES

CHANGE

Description List system FC ports.

This command returns the link status and the throughput of an FC port.

Syntax

```
config.fc.port.query [port=PORT[,PORT,...]] [switch=SWITCH] [node=NODE]
```

Arguments

port	Node and FC port index, e.g: N2FC5, or WWPN (a 16-digit hexadecimal number, optionally divided by colons to 8 groups of two digits each) (multiple values, separated by commas)
switch	Name of an existing fc switch
node	Node name

Example

```
config.fc.port.query name=N1FC1,N1FC2
```

Output

NAME	ENABLED	STATE	STATE DESCRIPTION	LINK	SWITCH	WWPN	ROLE	SPEED	MAX SPEED	PATCH PANEL
N1FC1	yes	OK	-	UP	Cisco1	57:42:b0:f0:00:0b:b8:11	HARD_PORT	8 Gbps	8 Gbps	1
N1FC2	yes	OK	-	DOWN	-	57:42:b0:f0:00:0b:b8:12	HARD_PORT	-	8 Gbps	2

Chapter 30. Config FC Switch

config.fc.switch.delete ADMIN

CHANGE

Description Delete FC switch.

Syntax

```
config.fc.switch.delete switch=SWITCH[,SWITCH,...]
```

Arguments

switch	Name of an existing fc switch (multiple values, separated by commas)

Example

```
config.fc.switch.delete name=brocade-1
```

Output

```
FC switch brocade-1 deleted
```

config.fc.switch.query ALL ROLES

CHANGE

Description List FC switches.

Lists the switches that are connected to the InfiniBox. The query can be filtered by switch name.

Syntax

```
config.fc.switch.query [switch=SWITCH[,SWITCH,...]]
```

Arguments

switch	Name of an existing fc switch (multiple values, separated by commas)

Example

```
config.fc.switch.query
```

Output

NAME	WWNN	RESILIENCY	VENDOR
Brocade1	10:00:50:eb:1a:2a:d3:54	Single point of failure	Brocade
Brocade6	10:00:50:eb:1a:2a:d3:55	Connected	Brocade
Brocade8	10:00:50:eb:1a:2a:d3:56	Disconnected	Brocade
Cisco7	20:01:54:7f:ee:d8:c4:c1	Disconnected	Cisco

config.fc.switch.rename ADMIN

CHANGE

Description Rename FC switch.

Renames the FC switch

Syntax

```
config.fc.switch.rename switch=SWITCH new_name=NAME
```

Arguments

new_name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
switch	Name of an existing fc switch

Example

```
config.fc.switch.rename name=brocade new_name=brocade1
```

Output

```
FC switch renamed from brocade1 to brocade-1
```

Chapter 31. Config Cache

Set SSD cache default for new pools.

Chapter 32. Config Notifications SMTP

SMTP commands allow you to configure an SMTP target for the InfiniBox events.

config.notifications.smtp.modify ADMIN

CHANGE

Description Modify SMTP notification target.

Set an SMTP target for InfiniBox notifications. An SMTP server address is required. Optionally, you can define a new server by setting the host, port, user and password. You can also determine whether to use the secured TLS protocol. You can determine a prefix that will be added to the subject of the email.

Syntax

```
config.notifications.smtp.modify smtp=SMTPTARGET [host=HOSTNAME] [port=PORT] [tls=YESNO]
[username=STR] [password=STR] [fromaddress=LOCALEMAIL] [subject_prefix=STR]
```

Arguments

subject_prefix	Any sequence of characters
fromaddress	a valid email address
password	Any sequence of characters
username	Any sequence of characters
tls	Either yes, or no
port	An integer number between 0 and 65535 (inclusive)
host	A valid hostname or IPv4 address
smtp	Name of an existing smtp notification target

Example

```
config.notifications.smtp.modify name=SMTP-1 host=smtp port=25 tls=tls fromaddress=no-
reply@infinidat.com
```

Output

```
SMTP notification target SMTP-1 modified
```

config.notifications.smtp.query ALL ROLES

CHANGE

Description List existing SMTP notification targets.

Syntax

```
config.notifications.smtp.query
```

Example

```
config.notifications.smtp.query
```

Output

NAME	HOST	PORT	TLS	USERNAME	FROM ADDRESS
customer-smtp	smtp	25	tls	-	no-reply@infinidat.com

config.notifications.smtp.rename ADMIN

CHANGE

Description Rename SMTP notification target.

Syntax

```
config.notifications.smtp.rename smtp=SMTPTARGET new_name=NAME
```

Arguments

new_name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
smtp	Name of an existing smtp notification target

Example

```
config.notifications.smtp.rename name=customer-smtp new_name=customer2-smtp
```

Output

```
SMTP notification target customer-smtp renamed to customer2-smtp
```

config.notifications.smtp.test ADMIN

CHANGE

Description Test SMTP notification target.

This command sends a test notification to an email recipient.

Syntax

```
config.notifications.smtp.test smtp=SMTPTARGET recipient=LOCALEMAIL
```

Arguments

smtp	Name of an existing smtp notification target
recipient	a valid email address

Example

```
config.notifications.smtp.test name=customer-smtp recipient=customer@customer.com
```

Output

```
SMTP notification target customer-smtp tested successfully
```

Chapter 33. Config Notifications SNMP

SNMP commands allow you to configure an SNMP trap for the InfiniBox events.

config.notifications.snmp.define ADMIN

CHANGE

Description Create a new SNMP notification target.

Define an SNMP trap to which the InfiniBox sends events. State the name and host and then select the version of the SNMP protocol.

Syntax

```
config.notifications.snmp.define name=NAME host=HOSTNAME version=SNMPPROTOCOL [port=PORT]
[community=STR] [engine_id=SNMPENGINE] [username=STR] [security=SNMPSECURITY] [password=STR]
[auth_type=SNMPAUTHPROTOCOL] [priv_key=STR] [encryption=SNMPENCRYPTION]
```

Arguments

`auth_type`	MD5 or SHA
`password`	Any sequence of characters
`security`	NoAuthNoPriv, AuthNoPriv or AuthPriv
`username`	Any sequence of characters
`engine_id`	A maximum of 12 characters. Starting woth 0x and is fallowed by letters and digits
`community`	Any sequence of characters
`port`	An integer number between 0 and 65535 (inclusive)
`version`	1, 2c or 3
`host`	A valid hostname or IPv4 address
`name`	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@() []$=!-#{}%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
`encryption`	AES or DES
`priv_key`	Any sequence of characters

The following example describes how to set an SNMP target for SNMP versions 1 and 2c. For these SNMP versions you need to define a community.

Example

```
config.notifications.snmp.define name=snmp-v1-dest host=snmp version=1 community=infinidat
```

Output

```
SNMP notification target snmp-v1-dest created
```

The following example describes how to set an SNMP target for SNMP version 3. For this SNMP version you need to define an SNMP Engine ID and a user name. You also need to select the user-based security model among authentication and privacy, authentication without privacy and no authentication and no privacy. Optionally, you may set the authentication type, and whether to encrypt the events and a private key.

Example

```
config.notifications.snmp.define name=snmp-v3-dest host=snmp version=3 engine_id=1 username=user1
security=NoAuthNoPriv
```

Output

```
SNMP notification target snmp-v3-dest created
```

In this SNMP version 3 example, the messages are authenticated and not encrypted.

Example

```
config.notifications.snmp.define name=snmp-v3-dest host=snmp version=3 engine_id=1 username=user1
security=AuthPriv password=123456 auth_type=SHA priv_key=privatekey encryption=DES
```

Output

```
SNMP notification target snmp-v3-dest created
```

In this SNMP version 3 example, the messages are both authenticated and encrypted.

Example

```
config.notifications.snmp.define name=snmp-v3-authpriv host=snmp version=3 engine_id=1
username=user1 security=AuthPriv password=123456 auth_type=SHA priv_key=privatekey encryption=DES
```

Output

```
SNMP notification target snmp-v3-dest created
```

config.notifications.snmp.modify ADMIN

CHANGE

Description **Modify SNMP notification target.**

Modify some of the SNMP definitions.

Syntax

```
config.notifications.snmp.modify snmp=SNMPTARGET [host=HOSTNAME] [port=PORT] [community=STR]
[engine_id=SNMPENGINE] [username=STR] [security=SNMPSECURITY] [password=STR]
[auth_type=SNMPAUTHPROTOCOL] [priv_key=STR] [encryption=SNMPENCRYPTION]
```

Arguments

security	NoAuthNoPriv, AuthNoPriv or AuthPriv
username	Any sequence of characters
engine_id	A maximum of 12 characters. Starting woth 0x and is fallowed by letters and digits
community	Any sequence of characters
port	An integer number between 0 and 65535 (inclusive)
host	A valid hostname or IPv4 address
snmp	Name of an existing snmp notification target
password	Any sequence of characters
auth_type	MD5 or SHA
encryption	AES or DES
priv_key	Any sequence of characters

Example

```
config.notifications.snmp.define name=NAME host=HOSTNAME version=SNMPPROTOCOL [port=PORT]
[community=STR]
```

Output

```
SNMP notification target snmp1 updated
```

config.notifications.snmp.query ALL ROLES

CHANGE

Description **List existing SNMP notification targets.**

Syntax

```
config.notifications.snmp.query [snmp=SNMPTARGET]
```

Arguments

snmp	Name of an existing snmp notification target

Example

```
config.notifications.snmp.query [name=SNMPTARGET]
```

Output

NAME	HOST	PORT	VERSION	COMMUNITY	ENGINE ID	USERNAME	SECURITY
snmpv1	snmp	25	SNMPv1	infinidat	-	-	-
snmpv2	snmp	25	SNMPv2c	infinidat	-	-	-
snmpv3	snmp	25	SNMPv3	-	1	user1	NoAuthNoPriv

config.notifications.snmp.remove ADMIN

CHANGE

Description Delete SNMP notification target.

Syntax

```
config.notifications.snmp.remove snmp=SNMPTARGET
```

Arguments

snmp	Name of an existing snmp notification target

Example

```
config.notifications.snmp.remove name=SNMPTARGET
```

Output

```
SNMP notification target snmp1 deleted
```

config.notifications.snmp.rename ADMIN

CHANGE

Description Rename SNMP notification target.

Syntax

```
config.notifications.snmp.rename snmp=SNMPTARGET new_name=NAME
```

Arguments

new_name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#{}%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
snmp	Name of an existing snmp notification target

Example

```
config.notifications.snmp.rename name=SNMPTARGET new_name=NAME
```

Output

```
SNMP notification target snmp1 renamed to snmp2
```

config.notifications.snmp.test ADMIN

CHANGE

Description Test SNMP notification target.

Syntax

```
config.notifications.snmp.test snmp=SNMPTARGET
```

Arguments

snmp	Name of an existing snmp notification target

Example

```
config.notifications.snmp.test name=SNMPTARGET
```

Output

```
SNMP notification target snmp1 tested successfully
```

Chapter 34. Config Notifications Syslog

Syslog commands allow you to log and store InfiniBox events.

config.notifications.syslog.define ADMIN

CHANGE

Description **Create a new syslog notification target.**

Define a syslog notification target by providing a host and a port. You can override the default UDP protocol with TCP. You can override the default Facility local7 with any other value between local0 and local7. Facility is a part of the Syslog message Priority and local0 contributes to the calculation that yields the highest priority.

Syntax

```
config.notifications.syslog.define name=NAME host=HOSTNAME [port=PORT] [protocol=SYSLOGPROTOCOL]
[facility=SYSLOGFACILITY]
```

Arguments

facility	local0, local1, local2, local3, local4, local5, local6 or local7
protocol	TCP or UDP
port	An integer number between 0 and 65535 (inclusive)
host	A valid hostname or IPv4 address
name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.

Example

```
config.notifications.syslog.define name=target0 host=syslog1
```

Output

```
Syslog notification target0 syslog1 created
```

config.notifications.syslog.modify ADMIN

CHANGE

Description **Modify syslog notification target.**

Syntax

```
config.notifications.syslog.modify syslog=SYSLOGTARGET [host=HOSTNAME] [port=PORT]
[protocol=SYSLOGPROTOCOL] [facility=SYSLOGFACILITY]
```

Arguments

protocol	TCP or UDP
port	An integer number between 0 and 65535 (inclusive)
host	A valid hostname or IPv4 address
syslog	Name of an existing syslog notification target
facility	local0, local1, local2, local3, local4, local5, local6 or local7

Example

```
config.notifications.syslog.modify name=target0 host=syslog2
```

Output

```
Syslog notification target0 syslog2 updated
```

config.notifications.syslog.query ALL ROLES

CHANGE

Description List existing syslog notification targets.

Syntax

```
config.notifications.syslog.query [syslog=SYSLOGTARGET]
```

Arguments

syslog	Name of an existing syslog notification target

Example

```
config.notifications.syslog.query
```

Output

NAME	HOST	PORT	PROTOCOL	FACILITY
target0	syslog	25	UDP	local0
target1	syslog	25	TCP	local1

config.notifications.syslog.remove ADMIN CHANGE

Description Delete syslog notification target.

Syntax

```
config.notifications.syslog.remove syslog=SYSLOGTARGET
```

Arguments

syslog	Name of an existing syslog notification target

Example

```
config.notifications.syslog.remove name=target1
```

Output

```
Syslog notification target target1 deleted
```

config.notifications.syslog.rename ADMIN CHANGE

Description Rename syslog notification target.

Syntax

```
config.notifications.syslog.rename syslog=SYSLOGTARGET new_name=NAME
```

Arguments

new_name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
syslog	Name of an existing syslog notification target

Example

```
config.notifications.syslog.rename name=SYSLOGTARGET new_name=NAME
```

Output

```
Syslog notification target target1 renamed to target2
```

config.notifications.syslog.test ADMIN CHANGE

Description Test syslog notification target.

Syntax

```
config.notifications.syslog.test syslog=SYSLOGTARGET
```

Arguments

syslog	Name of an existing syslog notification target

Example

```
config.notifications.syslog.test name=SYSLOGTARGET
```

Output

```
Syslog notification target target1 tested successfully
```

Chapter 35. Config LDAP

LDAP commands allow users listed on an LDAP to successfully login to an InfiniBox.

config.ldap.define ADMIN

CHANGE

Description Define an LDAP server.

Allow a user that is listed in an LDAP server to access the InfiniBox.

Syntax

```
config.ldap.define name=NAME username=STR type=LDAP type [domain=STR] [password=STR]
[server=STR[,STR,...]] [schema_user_class=STR] [port=PORT] [use_ldaps=YESNO]
[schema_username_attribute=STR] [schema_user_basedn=STR] [schema_group_class=STR]
[schema_group_attribute=STR] [schema_member_attribute=STR] [schema_group_basedn=STR]
```

Arguments

name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
username	Any sequence of characters
type	AD or LDAP
schema_user_basedn	Any sequence of characters
schema_username_attribute	Any sequence of characters
use_ldaps	Either yes, or no
port	An integer number between 0 and 65535 (inclusive)
schema_user_class	Any sequence of characters
server	Any sequence of characters (multiple values, separated by commas)
password	Any sequence of characters
domain	Any sequence of characters
schema_group_class	Any sequence of characters
schema_group_attribute	Any sequence of characters
schema_member_attribute	Any sequence of characters
schema_group_basedn	Any sequence of characters

Example

```
config.ldap.define name=ad2k3.local domain=ad2k3.local username=Administrator
```

Output

```
LDAP infinidat.com defined
```

config.ldap.flush_cache ADMIN

CHANGE

Description Flush LDAP authentication cache.

Clear the cache before you define new LDAP users.

Syntax

```
config.ldap.flush_cache
```

Example

```
config.ldap.flush_cache
```

Output

```
LDAP authentication cache flushed
```

config.ldap.modify ADMIN

CHANGE

Description Modify the LDAP server.

Change any of the LDAP server details.

Syntax

```
config.ldap.modify ldap=USERSREPOSITORY [username=STR] [password=STR] [server=STR[,STR,...]]
[schema_user_class=STR] [port=PORT] [use_ldaps=YESNO] [schema_username_attribute=STR]
[schema_user_basedn=STR] [schema_group_class=STR] [schema_group_attribute=STR]
[schema_member_attribute=STR] [schema_group_basedn=STR]
```

Arguments

schema_member_attribute	Any sequence of characters
schema_group_attribute	Any sequence of characters
schema_group_class	Any sequence of characters
schema_user_basedn	Any sequence of characters
schema_username_attribute	Any sequence of characters
use_ldaps	Either yes, or no
port	An integer number between 0 and 65535 (inclusive)
schema_user_class	Any sequence of characters
server	Any sequence of characters (multiple values, separated by commas)
password	Any sequence of characters
username	Any sequence of characters
ldap	Name of an existing ldap server
schema_group_basedn	Any sequence of characters

Modifying the user and password

Example

```
config.ldap.modify name=ad2k3.local domain=ad2k3.local username=Administrator
```

Output

```
LDAP ad2k3.local modified
```

Modifying the port

Example

```
config.ldap.modify name=ad2k3.local domain=ad2k3.local port=626
```

Output

```
LDAP ad2k3.local modified
```

config.ldap.order_query ALL ROLES

CHANGE

Description LDAP & Active Directory resolution order.

Syntax

```
config.ldap.order_query
```

Example

```
config.ldap.order_query
```

Output

INDEX	LDAP
1	LDAP-1
2	LDAP-2

config.ldap.query ALL ROLES

CHANGE

Description List existing LDAP servers.

List the LDAP user along with its attributes.

Syntax

```
config.ldap.query [ldap=USERSREPOSITORY] [type=LDAP type]
```

Arguments

ldap	Name of an existing ldap server
type	AD or LDAP

Example

```
config.ldap.query
```

Output

Name	ad2k3.local
Repository Type	ActiveDirectory
Domain Name	ad2k3.local
Bind Username	Administrator
User Class	user
Username Attribute	sAMAccountName
Users Basedn	-
Group Class	cn
Group Name Attribute	cn
Group Memberof Attribute	memberof
Groups Basedn	-

config.ldap.remove ADMIN

CHANGE

Description Remove the LDAP server definition.

Disconnect the LDAP server from the InfiniBox, so that users listed on this LDAP server can no longer access the InfiniBox.

Syntax

```
config.ldap.remove ldap=USERSREPOSITORY
```

Arguments

ldap	Name of an existing ldap server

Example

```
config.ldap.remove name=ACTIVEDIRECTORY
```

Output

```
LDAP infinidat.com removed
```

config.ldap.rename ADMIN

CHANGE

Description Rename the LDAP server.

Syntax

```
config.ldap.rename ldap=USERSREPOSITORY new_name=NAME
```

Arguments

new_name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@()[]$=!-#{}%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
ldap	Name of an existing ldap server

Example

```
config.ldap.rename name=ACTIVEDIRECTORY new_name=NAME
```

Output

```
LDAP infinidat.com renamed to customer.com
```

config.ldap.set_order ADMIN

CHANGE

Description Set LDAP servers resolution order.

Syntax

```
config.ldap.set_order order=USERSREPOSITORY[,USERSREPOSITORY,...]
```

Arguments

order	Name of an existing ldap server (multiple values, separated by commas)

Example

```
config.ldap.set_order order=ACTIVEDIRECTORY
```

Output

```
Order set
```

config.ldap.test ADMIN

CHANGE

Description Test the LDAP server.

Syntax

```
config.ldap.test ldap=USERSREPOSITORY
```

Arguments

ldap	Name of an existing ldap server

Example

```
config.ldap.test name=ACTIVEDIRECTORY
```

Output

```
LDAP tested successfully
```

Chapter 36. Config Ethernet Port

config.ethernet.port.query ALL ROLES

CHANGE

Description List system ethernet ports.

The port speed is in Gbps

Syntax

```
config.ethernet.port.query [port=PORT] [node=NODE]
```

Arguments

port	Ethernet port index, e.g: ETH2
node	Node name

Run the command with a detailed output

Example

```
config.ethernet.port.query --detailed
```

Output

NAME	ROLE	STATE	LINK	CURRENT SPEED	MAX SPEED	PATCH PANEL	HW ADDR	DEVICE
N1ETH1	Data	OK	UP	10 Gbps	10 Gbps	9	ec:f4:bb:e0:87:60	eth-data1
N1ETH2	Data	OK	UP	-	10 Gbps	10	ec:f4:bb:e0:87:60	eth-data2
...								

Run the command with no parameters

Example

```
config.ethernet.port.query
```

Output

NAME	ROLE	STATE	LINK	CURRENT SPEED	MAX SPEED	PATCH PANEL
N1ETH1	Data	OK	UP	10 Gbps	10 Gbps	9
N1ETH2	Data	OK	UP	-	10 Gbps	10
N1ETH3	Management	OK	UP	-	1 Gbps	MGMT
N1ETH4	Internal	OK	UP	-	1 Gbps	-
N1ETH5	iDrac	OK	UP	1 Gbps	1 Gbps	-
N2ETH1	Data	OK	UP	10 Gbps	10 Gbps	9
N2ETH2	Data	OK	UP	-	10 Gbps	10
N2ETH3	Management	OK	UP	-	1 Gbps	MGMT
N2ETH4	Internal	OK	UP	-	1 Gbps	-
N2ETH5	iDrac	OK	UP	1 Gbps	1 Gbps	-
N3ETH1	Data	OK	UP	10 Gbps	10 Gbps	9
N3ETH2	Data	OK	UP	-	10 Gbps	10
N3ETH3	Management	OK	UP	-	1 Gbps	MGMT
N3ETH4	Internal	OK	UP	-	1 Gbps	-
N3ETH5	iDrac	OK	UP	1 Gbps	1 Gbps	-

Chapter 37. Config Ethernet Interface

config.ethernet.interface.add_port ADMIN

CHANGE

Description Add ethernet port to a network interface.

You can add several ports at once. The command returns success and error messages per port.

Syntax

```
config.ethernet.interface.add_port interface=PHYSICALINTERFACE port=PORT[,PORT,...] [node=NODE]
```

Arguments

interface	Name of an existing physical network interface
port	Ethernet port index, e.g: ETH2 (multiple values, separated by commas)
node	Node name

Example

```
config.ethernet.interface.add_port name=interface port=N1ETH1
```

Output

```
Port added: N1ETH1
```

config.ethernet.interface.create ADMIN CHANGE

Description Create a network interface.

Syntax

```
config.ethernet.interface.create ports=PORT[,PORT,...] name=NETWORKINTERFACENAME [type=TYPE]
[node=NODE]
```

Arguments

ports	Ethernet port index, e.g: ETH2 (multiple values, separated by commas)
type	Only one value: PORT_GROUP
name	A maximum of 5 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
node	Node name

Example

```
config.ethernet.interface.create name=interface1 type=PORT_GROUP ports=N1ETH3
repeat_on_all_nodes=yes
```

Output

```
interface interface1: created on node_1
interface interface1: created on node_2
interface interface1: created on node_3
```

config.ethernet.interface.create_vlan ADMIN CHANGE

Description Create network interface of type VLAN

Syntax

```
config.ethernet.interface.create_vlan physical_interface=PHYSICALINTERFACE vlan_id=INT [node=NODE]
```

Arguments

physical_interface	Name of an existing physical network interface
node	Node name
vlan_id	An integer number

Example

```
config.ethernet.interface.create_vlan physical_interface=PG1 vlan_id=400 node=node-1
```

Output

```
VLAN PG1.400 created on interface PG1 (node-1)
```

config.ethernet.interface.delete ADMIN

CHANGE

Description Delete network interface.

The interface has to be diabled in order for this command to take effect.

Syntax

```
config.ethernet.interface.delete interface=INTERFACE [node=NODE]
```

Arguments

interface	Name of an existing network interface
node	Node name

Example

```
config.ethernet.interface.delete name=interface1
```

Output

```
Network interface n1eth deleted
```

config.ethernet.interface.disable ADMIN

CHANGE

Description Disable network interface.

Syntax

```
config.ethernet.interface.disable interface=INTERFACE [node=NODE]
```

Arguments

interface	Name of an existing network interface
node	Node name

Example

```
config.ethernet.interface.disable name=interface1
```

Output

```
Network interface n1eth disabled
```

config.ethernet.interface.enable ADMIN

CHANGE

Description Enable network interface.

Syntax

```
config.ethernet.interface.enable interface=PHYSICALINTERFACE [node=NODE]
```

Arguments

interface	Name of an existing physical network interface
node	Node name

Example

```
config.ethernet.interface.enable name=interface1
```

Output

```
Network interface n1eth enabled
```

config.ethernet.interface.query ALL ROLES

CHANGE

Description List existing network interfaces.

Syntax

```
config.ethernet.interface.query [interface=INTERFACE[,INTERFACE,...]] [node=NODE]
```

Arguments

interface	Name of an existing network interface (multiple values, separated by commas)
node	Node name

Example

```
config.ethernet.interface.query
```

Output

NAME	TYPE	NODE	PORTS	NETWORK SPACES	STATE
n1eth-data1	PORT	1	N1ETH1	ns1	OK
n2eth-data1	PORT	2	N2ETH1	ns1	OK
n3eth-data1	PORT	3	N3ETH1	ns1	OK

config.ethernet.interface.remove_port ADMIN

CHANGE

Description Remove ethernet port from network interface.

Syntax

```
config.ethernet.interface.remove_port interface=PHYSICALINTERFACE port=PORT[,PORT,...] [node=NODE]
```

Arguments

interface	Name of an existing physical network interface
port	Ethernet port index, e.g: ETH2 (multiple values, separated by commas)
node	Node name

Example

```
config.ethernet.interface.remove_port name=interface1 port=N1ETH1
```

Output

```
Port removed: N1ETH1
```

config.ethernet.interface.rename ADMIN

CHANGE

Description Rename network interface.

Syntax

```
config.ethernet.interface.rename interface=PHYSICALINTERFACE new_name=NETWORKINTERFACENAME
[node=NODE]
```

Arguments

new_name	A maximum of 5 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
interface	Name of an existing physical network interface
node	Node name

Example

```
config.ethernet.interface.rename name=interface1 new_name=N1ETH1
```

Output

```
Interface n1eth renamed to n1eth1
```

Chapter 38. Network space

config.net_space.create ADMIN

CHANGE

Description Create network space.

Syntax

```
config.net_space.create name=NAME service=SERVICE interface=INTERFACE[,INTERFACE,...]
network=IPADDRESS/CIDR [default_gateway=IPADDRESS] [mtu=MTU] [rate_limit=LIMIT]
[replication_type=NETWORKRMRTYPE]
```

Arguments

network	A valid IPv4 network address in CIDR notation, e.g. 192.168.1.0/24
interface	Name of an existing network interface (multiple values, separated by commas)
service	NAS, ISCSI or REPLICATION
name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#{}%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
mtu	Maximum transmission unit. An integer number between 1280 and 9000 (inclusive)
default_gateway	A valid IPv4 address
rate_limit	NONE, or an amount of megabits per second
replication_type	Network space replication type

Example

```
config.net_space.create name=netspace1 interface=n1eth-data1 default_gateway=172.16.95.254
service=REPLICATION network=172.16.64.0/19
```

Output

```
Net_space netspace1 created
```

config.net_space.delete ADMIN

CHANGE

Description Delete network space.

Syntax

```
config.net_space.delete net_space=NETWORKSPACE
```

Arguments

net_space	Name of an existing network space

Example

```
config.net_space.delete name=netspace1
```

Output

```
Net_space netspace1 deleted
```

config.net_space.params.query ALL ROLES

CHANGE

Description List network space service parameters.

Syntax

```
config.net_space.params.query [net_space=NETWORKSPACE] [service=SERVICE]
```

Arguments

net_space	Name of an existing network space
service	NAS, ISCSI or REPLICATION

Example

```
config.net_space.params.query net_space=iscsi-data1
```

Output

NAME	IQN	TCP PORT	UNKNOWN HOST	AUTH TYPE	ISNS SERVERS
iscsi-data1	iqn.2009-11.com.infinidat:storage:infinibox-sn-3009	3260	None		172.16.64.96

config.net_space.params.set_iscsi

ADMIN POOL ADMIN

CHANGE

Description Set network space iSCSI parameters.

Syntax

```
config.net_space.params.set_iscsi net_space=NETWORKSPACE [tcp_port=TCPPORT]
[isns_servers=HOSTNAME[,HOSTNAME,...]]
```

Arguments

isns_servers	NONE, or a valid hostname or IPv4 address (multiple values, separated by commas)
tcp_port	An integer number between 1 and 65535 (inclusive)
net_space	Name of an existing network space

Example

```
config.net_space.params.set_iscsi net_space=iscsi-data1 tcp_port=3260
```

Output

```
Network space "iscsi-data1" updated
```

config.net_space.query

ALL ROLES

CHANGE

Description List existing network spaces.

Syntax

```
config.net_space.query [net_space=NETWORKSPACE] [service=SERVICE]
```

Arguments

net_space	Name of an existing network space
service	NAS, ISCSI or REPLICATION

Non-filtered query

Example

```
config.net_space.query
```

Output

NAME	SERVICE	NETWORK	MTU	RATE LIMIT	GATEWAY	IPS	INTERFACES
rmr_ns	REPLICATION	172.16.32.0/19	1500	1000 Mbps	-	7	LAG-1,LAG-2,LAG-3

Filtered by service query.

Example

```
config.net_space.query service=NAS
```

Output

NAME	SERVICE	NETWORK	MTU	RATE LIMIT	GATEWAY	IPS	INTERFACES
default_nas_space	NAS	172.16.32.0/19	9000	-	-	6	n1eth-data1,n2eth-data1,n3eth-data1

config.net_space.rename ADMIN

CHANGE

Description Rename network space.

Syntax

```
config.net_space.rename net_space=NETWORKSPACE new_name=NAME
```

Arguments

new_name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@() []$=!-#{}%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
net_space	Name of an existing network space

Example

```
config.net_space.rename name=rmr_ns new_name=rmr_ns_1
```

Output

```
Network space rmr_ns renamed to rmr_ns_1
```

config.net_space.route.add ADMIN

CHANGE

Description **Add route to network space.** The routing table is automatically created by InfiniBox to each network space that has a default gateway definition.

Syntax

```
config.net_space.route.add net_space=NETWORKSPACE destination=IPADDRESS netmask=NETMASK
gateway=IPADDRESS
```

Arguments

gateway	A valid IPv4 address
netmask	An IPv4 netmask
destination	A valid IPv4 address
net_space	Name of an existing network space

Example

```
config.net_space.route.add name=Replication destination=192.168.10.0 netmask=255.255.255.0
gateway=10.68.65.254 metric=20
```

Output

```
Route added to Network Space Replication
```

config.net_space.route.query ALL ROLES

CHANGE

Description Query network space routes.

Syntax

```
config.net_space.route.query [net_space=NETWORKSPACE] [destination=IPADDRESS]
```

Arguments

destination	A valid IPv4 address
net_space	Name of an existing network space

Example

```
config.net_space.route.query --detailed
```

Output

Name	Replication
Service	SYNC Replication
Network	172.16.32.0/19
MTU	1500
Rate Limit	-
Gateway	172.16.63.254
IPs	6
Interfaces	PG1
VMAC Addresses	address : role 74:2b:0f:4e:56:f8 : DATA 74:2b:0f:4e:56:f9 : DATA 74:2b:0f:4e:56:f4 : DATA 74:2b:0f:4e:56:f5 : DATA 74:2b:0f:4e:56:f6 : DATA 74:2b:0f:4e:56:f7 : DATA

Example

```
config.net_space.route.query net_space=Replication
```

Output

Destination	Netmask	Gateway
0.0.0.0	0.0.0.0	172.25.172.254
10.0.0.192	255.255.255.240	172.25.172.1
192.168.0.0	255.255.0.0	172.25.172.1
172.25.172.255	172.25.172.3	172.25.172.254

config.net_space.route.remove ADMIN CHANGE

Description Remove network space route.

Syntax

```
config.net_space.route.remove net_space=NETWORKSPACE destination=IPADDRESS netmask=NETMASK
gateway=IPADDRESS
```

Arguments

gateway	A valid IPv4 address
net_space	Name of an existing network space
destination	A valid IPv4 address
netmask	An IPv4 netmask

Example

```
config.net_space.route.remove net_space=Async1 destination=11.11.128.0 netmask=255.255.252.0
gateway=172.16.95.254
```

Output

```
Route to 11.11.128.0 255.255.252.0 removed from network space Async1
```

config.net_space.route.set_gateway ADMIN

CHANGE

Description **Set route gateway in network space.** This command allows to change the gateway that the route uses. The route is identified by all of its attributes.

Syntax

```
config.net_space.route.set_gateway net_space=NETWORKSPACE destination=IPADDRESS netmask=NETMASK gateway=IPADDRESS new_gateway=IPADDRESS
```

Arguments

net_space	Name of an existing network space
destination	A valid IPv4 address
netmask	An IPv4 netmask
gateway	A valid IPv4 address
new_gateway	A valid IPv4 address

Example

```
config.net_space.route.set_gateway net_space=Replication netmask=255.255.255.240
destination=192.168.0.0 gateway=172.25.172.1 new_gateway=172.25.172.2
```

Output

```
Route was updated
```

config.net_space.set_default_gateway ADMIN

CHANGE

Description Set default gateway.

Syntax

```
config.net_space.set_default_gateway net_space=NETWORKSPACE default_gateway=IPADDRESSORNONE
```

Arguments

net_space	Name of an existing network space
default_gateway	A valid IPv4 address

Setting a default gateway.

Example

```
config.net_space.set_default_gateway name=REPLICATION default_gateway=172.16.32.254
```

Output

```
Network space REPLICATION default gateway set to 172.16.32.254
```

Removing the default gateway.

Example

```
config.net_space.set_default_gateway name=REPLICATION default_gateway=None
```

Output

```
Network space REPLICATION default gateway unset
```

config.net_space.set_interfaces ADMIN

CHANGE

Description Set network space port groups.

Syntax

```
config.net_space.set_interfaces net_space=NETWORKSPACE interfaces=INTERFACE[,INTERFACE,...]
```

Arguments

interfaces	Name of an existing network interface (multiple values, separated by commas)
net_space	Name of an existing network space

Example

```
config.net_space.set_interfaces name=REPLICATION interfaces=n1eth-data1, n2eth-data1, n3eth-data1
```

Output

```
Network space REPLICATION interfaces set to n1eth-data1, n2eth-data1, n3eth-data1
```

config.net_space.set_mtu ADMIN

CHANGE

Description Set the network space MTU.

MTU (maximum transfer unit) is measured in octets (bytes).

Syntax

```
config.net_space.set_mtu net_space=NETWORKSPACE mtu=MTU
```

Arguments

mtu	Maximum transmission unit. An integer number between 1280 and 9000 (inclusive)
net_space	Name of an existing network space

Example

```
config.net_space.set_mtu name=RMR01 mtu=1200
```

Output

```
Network space RMR01 MTU set to 1200
```

config.net_space.set_network ADMIN

CHANGE

Description Set network space network configuration.

Syntax

```
config.net_space.set_network net_space=NETWORKSPACE network=IPADDRESS/CIDR
```

Arguments

network	A valid IPv4 network address in CIDR notation, e.g. 192.168.1.0/24
net_space	Name of an existing network space

Example

```
config.net_space.set_network name=REPLICATION network=172.16.32.0/19 default_gateway=none
```

Output

```
Network space REPLICATION network set to 172.16.32.0/19
```

config.net_space.set_rate_limit ADMIN POOL ADMIN

CHANGE

Description Sets a bandwidth limitation on a Network Space per node.

This limit is configured on the Interface used by the Network space to allow traffic shaping and avoid bandwidth starvation.

Syntax

```
config.net_space.set_rate_limit net_space=NETWORKSPACE limit_per_node=LIMIT
```

Arguments

limit_per_node	NONE, or an amount of megabits per second
net_space	Name of an existing network space

Setting a rate limit.

Example

```
config.net_space.set_rate_limit name=replication_space limit_per_node=200
```

Output

```
Network space replication_space bandwidth limitation (per node): Limited to 200Mbps
```

Removing the rate limit.

Example

```
config.net_space.set_rate_limit name=replication_space limit_per_node=none
```

Output

```
Network space replication_space bandwidth limitation (per node): Limit removed
```

Chapter 39. Network space IP addresses

config.net_space.ip.create ADMIN

CHANGE

Description Create IP address in network space.

Syntax

```
config.net_space.ip.create net_space=NETWORKSPACE ip_address=IPADDRESS[,IPADDRESS,...]
```

Arguments

ip_address	A valid IPv4 address (multiple values, separated by commas)
net_space	Name of an existing network space

Example

```
config.net_space.ip.create net_space=default.nas.space ip_address=172.16.32.56
```

Output

```
IP address 172.16.32.56 created in network space default_nas_space
```

config.net_space.ip.delete ADMIN

CHANGE

Description Delete IP address from network space.

The IP address has to be disabled in order for the deletion command to take effect.

Syntax

```
config.net_space.ip.delete net_space=NETWORKSPACE ip_address=IPADDRESS[,IPADDRESS,...]
```

Arguments

ip_address	A valid IPv4 address (multiple values, separated by commas)
net_space	Name of an existing network space

Example

```
config.net_space.ip.delete net_space=NETWORKSPACE ip_address=IPADDRESS
```

Output

```
IP address 172.16.32.57 deleted in network space default_nas_space
```

config.net_space.ip.disable ADMIN

CHANGE

Description Disable network space IP address.

Syntax

```
config.net_space.ip.disable net_space=NETWORKSPACE ip_address=IPADDRESS[,IPADDRESS,...]
```

Arguments

ip_address	A valid IPv4 address (multiple values, separated by commas)
net_space	Name of an existing network space

Example

```
config.net_space.ip.disable net_space=NETWORKSPACE ip_address=IPADDRESS
```

Output

```
IP address 172.16.32.57 disabled in network space default_nas_space
```

config.net_space.ip.enable ADMIN

CHANGE

Description Enable network space IP address.

Syntax

```
config.net_space.ip.enable net_space=NETWORKSPACE ip_address=IPADDRESS[,IPADDRESS,...]
```

Arguments

ip_address	A valid IPv4 address (multiple values, separated by commas)
net_space	Name of an existing network space

Example

```
config.net_space.ip.enable net_space=NETWORKSPACE ip_address=IPADDRESS
```

Output

```
IP address 172.16.32.57 enabled in network space default_nas_space
```

config.net_space.ip.query ALL ROLES

CHANGE

Description Query network space IPs.

Syntax

```
config.net_space.ip.query [net_space=NETWORKSPACE] [interface=INTERFACE]
```

Arguments

net_space	Name of an existing network space
interface	Name of an existing network interface

Example

```
config.net_space.ip.query
```

Output

NETWORK SPACE	IP ADDRESS	ENABLED	NODE	NETWORK INTERFACE	REPLICATION MANAGEMENT IP
RMR	172.16.32.162	yes	-	-	yes
RMR	172.16.32.163	yes	2	LAG-2	no
RMR	172.16.32.164	yes	2	LAG-2	no
RMR	172.16.32.165	yes	3	LAG-3	no
RMR	172.16.32.166	yes	3	LAG-3	no
RMR	172.16.32.167	yes	2	LAG-2	no
RMR	172.16.32.168	yes	3	LAG-3	no

Chapter 40. Filesystem metrics

metrics.nas.destination_ip.show ALL ROLES

CHANGE

Description Show metrics for specific destination ip in system.

Syntax

```
metrics.nas.destination_ip.show destination_ip=IPADDRESS [pool=POOL] [fs={FS|SNAP}]
[interval=Metrics Interval] [count=Metrics Count]
```

Arguments

interval	Metrics refresh interval. Max value is 60
fs	Name of an existing filesystem or snapshot
pool	Name of an existing pool
destination_ip	A valid IPv4 address
count	The number of times command will be executed

Example

```
metrics.nas.destination_ip.show destination_ip=172.16.32.0
```

Output

---	OPERATIONS	THROUGHPUT	LATENCY
READ	0	-	1.022 ms
WRITE	0	-	0.796 ms

metrics.nas.destination_ip.top ALL ROLES

CHANGE

Description Show top metrics for destination ips in system.

Syntax

```
metrics.nas.destination_ip.top [pool=POOL] [fs={FS|SNAP}] [sort_by=TYPE] [results=Metrics Results]
```

Arguments

fs	Name of an existing filesystem or snapshot
pool	Name of an existing pool
results	The number of results that will be displayed. Max value is 100
sort_by	ops, throughput or latency

Run the command with no parameters to see the metrics.

Example

```
metrics.nas.destination_ip.top
```

Output

DESTINATION IP	OPS	THROUGHPUT	LATENCY
10.77.255.34	17,895	-	0.281 ms
10.77.255.35	17,833	-	0.246 ms
10.77.255.31	17,828	-	0.274 ms
10.77.255.36	17,089	-	0.275 ms
10.77.255.32	17,634	-	0.258 ms
10.77.255.33	17,947	-	0.260 ms

metrics.nas.file.top ALL ROLES

CHANGE

Description Show top metrics for file paths in system.

Syntax

```
metrics.nas.file.top [pool=POOL] [fs={FS|SNAP}] [sort_by=TYPE] [results=Metrics Results]
```

Arguments

sort_by	ops, throughput or latency
fs	Name of an existing filesystem or snapshot
pool	Name of an existing pool
results	The number of results that will be displayed. Max value is 100

Example

```
metrics.nas.file.top
```

Output

File path	OPERATIOS	THROUGHPUT	LATENCY
/filesystem1/1.txt	290	-	1.022 ms
/fs_VMware/abcd.vmdk	314	-	0.796 ms
/folder1/folder2/10.txt	538	-	0.508 ms

metrics.nas.fs.show ALL ROLES

CHANGE

Description Show metrics for specific filesystem in system.

- The output can be filtered by filesystem
- The output can be configured to be refreshed every specific interval and count

Press Q to return to the command prompt.

Syntax

```
metrics.nas.fs.show fs={FS|SNAP} [interval=Metrics Interval] [count=Metrics Count]
```

Arguments

fs	Name of an existing filesystem or snapshot
count	The number of times command will be executed
interval	Metrics refresh interval. Max value is 60

Example

```
metrics.nas.fs.show fs=fs1
```

Output

---	OPERATIONS	THROUGHPUT	LATENCY
READ	0.0 ms	0	-
WRITE	0.0 ms	0	-

metrics.nas.fs.top ALL ROLES

CHANGE

Description Show top metrics for filesystems in system.

- The output can be filtered by pool
- The output can be sorted by any of the metrics (OPS, throughput, latency)
- The number of returned objects can be limited

Press Q to return to the command prompt.

Syntax

```
metrics.nas.fs.top [pool=POOL] [sort_by=TYPE] [results=Metrics Results]
```

Arguments

pool	Name of an existing pool
results	The number of results that will be displayed. Max value is 100
sort_by	ops, throughput or latency

Example

```
metrics.nas.fs.top
```

Output

FS ID	OPS	THROUGHPUT	LATENCY
44	290	-	1.022 ms
26	314	-	0.796 ms
31	538	-	0.508 ms
74	312	-	1.299 ms

metrics.nas.gid.show ALL ROLES

CHANGE

Description Show metrics for specific gid in system.

Syntax

```
metrics.nas.gid.show gid=INT [fs={FS|SNAP}] [interval=Metrics Interval] [count=Metrics Count]
```

Arguments

fs	Name of an existing filesystem or snapshot
gid	An integer number
count	The number of times command will be executed
interval	Metrics refresh interval. Max value is 60

Example

```
metrics.nas.gid.show
```

Output

---	OPERATIONS	THROUGHPUT	LATENCY
READ	0	-	1.022 ms
WRITE	0	-	0.796 ms

metrics.nas.gid.top ALL ROLES

CHANGE

Description Show top metrics for gid in system.

Syntax

```
metrics.nas.gid.top [pool=POOL] [fs={FS|SNAP}] [sort_by=TYPE] [results=Metrics Results]
```

Arguments

results	The number of results that will be displayed. Max value is 100
sort_by	ops, throughput or latency
fs	Name of an existing filesystem or snapshot
pool	Name of an existing pool

Example

```
metrics.nas.gid.top
```

Output

GROUP ID	OPERATIONS	THROUGHPUT	LATENCY
7184	1	-	0.727 ms
8926	1	-	0.744 ms
4582	1	-	0.508 ms
3473	1	-	1.299 ms

metrics.nas.pool.show ALL ROLES

CHANGE

Description Show metrics for specific pool in system.

Syntax

```
metrics.nas.pool.show pool=POOL [interval=Metrics Interval] [count=Metrics Count]
```

Arguments

interval	Metrics refresh interval. Max value is 60
pool	Name of an existing pool
count	The number of times command will be executed

Example

```
metrics.nas.pool.show pool=p1
```

Output

---	OPERATIONS	THROUGHPUT	LATENCY
READ	0	-	1.022 ms
WRITE	0	-	0.796 ms
METADATA	538	-	0.508 ms

metrics.nas.pool.top ALL ROLES

CHANGE

Description Show top metrics for pools in system.

Syntax

```
metrics.nas.pool.top [sort_by=TYPE] [results=Metrics Results]
```

Arguments

results	The number of results that will be displayed. Max value is 100
sort_by	ops, throughput or latency

Example

```
metrics.nas.pool.top
```

Output

POOL NAME	OPERATIONS	THROUGHPUT	LATENCY
p1	290	-	1.022 ms
p2	314	-	0.796 ms
p3	538	-	0.508 ms
p4	312	-	1.299 ms

metrics.nas.source_ip.show ALL ROLES

CHANGE

Description Show metrics for specific source ip in system.

Syntax

```
metrics.nas.source_ip.show source_ip=IPADDRESS [pool=POOL] [fs={FS|SNAP}] [interval=Metrics Interval] [count=Metrics Count]
```

Arguments

count	The number of times command will be executed
interval	Metrics refresh interval. Max value is 60
fs	Name of an existing filesystem or snapshot
pool	Name of an existing pool
source_ip	A valid IPv4 address

Example

```
metrics.nas.source_ip.show source_ip=172.32.16.0
```

Output

---	OPERATIONS	THROUGHPUT	LATENCY
READ	0	-	1.022 ms
WRITE	0	-	0.796 ms

metrics.nas.source_ip.top ALL ROLES

CHANGE

Description Show top metrics for the source IP addresses of a network space that serves NAS.

Syntax

```
metrics.nas.source_ip.top [pool=POOL] [fs={FS|SNAP}] [sort_by=TYPE] [results=Metrics Results]
```

Arguments

pool	Name of an existing pool
fs	Name of an existing filesystem or snapshot
results	The number of results that will be displayed. Max value is 100
sort_by	ops, throughput or latency

Example

```
metrics.nas.source_ip.top
```

Output

SOURCE IP	OPERATIONS	THROUGHPUT	LATENCY
44	290	-	1.022 ms
26	314	-	0.796 ms
31	538	-	0.508 ms
74	312	-	1.299 ms

metrics.nas.system.show ALL ROLES

CHANGE

Description Show metrics for system.

Syntax

```
metrics.nas.system.show [interval=Metrics Interval] [count=Metrics Count]
```

Arguments

count	The number of times command will be executed
interval	Metrics refresh interval. Max value is 60

Example

```
metrics.nas.system.show
```

Output

---	OPERATIONS	THROUGHPUT	LATENCY
READ	0	-	1.022 ms
WRITE	0	-	0.796 ms
METADATA	538	-	0.508 ms

metrics.nas.uid.show ALL ROLES

CHANGE

Description Show metrics for specific uid in system.

Syntax

```
metrics.nas.uid.show uid=INT [fs={FS|SNAP}] [interval=Metrics Interval] [count=Metrics Count]
```

Arguments

fs	Name of an existing filesystem or snapshot
uid	An integer number
interval	Metrics refresh interval. Max value is 60
count	The number of times command will be executed

Example

```
metrics.nas.uid.show uid=5
```

Output

---	OPERATIONS	THROUGHPUT	LATENCY
READ	0	-	1.022 ms
WRITE	0	-	0.796 ms

metrics.nas.uid.top ALL ROLES

CHANGE

Description Show top metrics for the system users.

Syntax

```
metrics.nas.uid.top [pool=POOL] [fs={FS|SNAP}] [sort_by=TYPE] [results=Metrics Results]
```

Arguments

results	The number of results that will be displayed. Max value is 100
sort_by	ops, throughput or latency
fs	Name of an existing filesystem or snapshot
pool	Name of an existing pool

Example

```
metrics.nas.uid.top
```

Output

USER ID	OPERATIONS	THROUGHPUT	LATENCY
44	290	-	1.022 ms
26	314	-	0.796 ms
31	538	-	0.508 ms
74	312	-	1.299 ms

Chapter 41. Replica metrics

These commands display metrics for a replicated volume or for the entire system.

metrics.async.fs.show ALL ROLES

CHANGE

Description Show metrics for specific replicating filesystem in system.

Syntax

```
metrics.async.fs.show fs={FS|SNAP} [interval=Metrics Interval] [count=Metrics Count]
```

Arguments

fs	Name of an existing filesystem or snapshot
interval	Metrics refresh interval. Max value is 60
count	The number of times command will be executed

Example

```
metrics.async.fs.show fs=fs1
```

Output

---	OPERATIONS	THROUGHPUT
received	2,337	813 MB/sec
sent	0	-

metrics.async.fs.top ALL ROLES

CHANGE

Description Show top metrics for asynchronously replicating filesystems in the system

Syntax

```
metrics.async.fs.top [results=Metrics Results]
```

Arguments

results	The number of results that will be displayed. Max value is 100

Example

```
metrics.async.fs.top
```

Output

FILESYSTEM NAME	OPERATIONS	THROUGHPUT
fs1	2,337	813 MB/sec
fs2	1,093	407 MB/sec

metrics.async.system.show ALL ROLES

CHANGE

Description Show metrics for replicated volumes in the system.

- Received - replication traffic that received by target volumes
- Sent - replication traffic that sent from source volumes

Syntax

```
metrics.async.system.show [interval=Metrics Interval] [count=Metrics Count]
```

Arguments

count	The number of times command will be executed
interval	Metrics refresh interval. Max value is 60

Example

```
metrics.async.system.show
```

Output

---	OPERATIONS	THROUGHPUT	LATENCY
received	2,337	813 MB/sec	0.851 ms
sent	0	-	0.0. ms

metrics.async.vol.show ALL ROLES

CHANGE

Description Show metrics for a specific replicated volume.

Syntax

```
metrics.async.vol.show vol={VOL|SNAP} [interval=Metrics Interval] [count=Metrics Count]
```

Arguments

count	The number of times command will be executed
interval	Metrics refresh interval. Max value is 60
vol	Name of an existing volume or snapshot

Example

```
metrics.async.vol.show vol=v1
```

Output

---	OPERATIONS	THROUGHPUT
received	2,337	813 MB/sec
sent	0	-

metrics.async.vol.top ALL ROLES

CHANGE

Description Show top ASYNC metrics for volumes in system.

Syntax

```
metrics.async.vol.top [results=Metrics Results]
```

Arguments

results	The number of results that will be displayed. Max value is 100

Example

```
metrics.async.vol.top
```

Output

VOL NAME	OPERATIONS	THROUGHPUT
v1	2,337	813 MB/sec
v2	1,093	407 MB/sec

Chapter 42. Volume Metrics

metrics.san.fc.show ALL ROLES

CHANGE

Description Show metrics for FC in system.

- The output can be filtered by pool
- The output can be filtered by volume
- The output can be configured to be refreshed every specified interval and count

Syntax

```
metrics.san.fc.show [pool=POOL] [vol={VOL|SNAP}] [interval=Metrics Interval] [count=Metrics Count]
```

Arguments

count	The number of times command will be executed
interval	Metrics refresh interval. Max value is 60
vol	Name of an existing volume or snapshot
pool	Name of an existing pool

Example

```
metrics.san.fc.show
```

Output

---	OPERATIONS	THROUGHPUT	LATENCY	ERRORS
READ	7,121	251 MB/sec	3663.017ms	0
WRITE	3,927	126 MB/sec	2803.351ms	0
XCOPY	0	-	0.0ms	0
WRITE_SAME	0	-	0.0ms	0

metrics.san.fc_initiator.show ALL ROLES

CHANGE

Description Show metrics for specific FC initiator in system.

Syntax

```
metrics.san.fc_initiator.show initiator=USEDFCLOGGEDININITIATOR [interval=Metrics Interval]
[count=Metrics Count]
```

Arguments

count	The number of times command will be executed
interval	Metrics refresh interval. Max value is 60
initiator	Logged in initiator

Example

```
metrics.san.fc_initiator.show initiator=10000000c9cc1979
```

Output

---	OPERATIONS	THROUGHPUT	LATENCY	ERRORS
READ	7,121	251 MB/sec	3663.017ms	0
WRITE	3,927	126 MB/sec	2803.351ms	0
XCOPY	0	-	0.0ms	0
WRITE_SAME	0	-	0.0ms	0

metrics.san.fc_initiator.top

ALL ROLES

CHANGE

Description Show metrics for top FC initiators in system.

Syntax

```
metrics.san.fc_initiator.top [pool=POOL] [host=HOST] [sort_by=TYPE] [results=Metrics Results]
```

Arguments

sort_by	operations, latency or throughput
host	Name of an existing host
pool	Name of an existing pool
results	The number of results that will be displayed. Max value is 100

Example

```
metrics.san.fc_initiator.top
```

Output

INITIATOR	OPERATIONS	THROUGHPUT	LATENCY	ERRORS
10:00:00:00:c9:ac:c6:f3	5,680	106 MB/sec	7.91 ms	0
10:00:00:00:c9:f2:1d:d9	4,790	313 MB/sec	8.12 ms	0

metrics.san.fc_target.show

ALL ROLES

CHANGE

Description Show metrics for specific port in system.

- The target is represented by the Port parameter

Syntax

```
metrics.san.fc_target.show port=PORT [interval=Metrics Interval] [count=Metrics Count]
```

Arguments

count	The number of times command will be executed
interval	Metrics refresh interval. Max value is 60
port	Node and FC port index, e.g: N2FC5, or WWPN (a 16-digit hexadecimal number, optionally divided by colons to 8 groups of two digits each)

Example

```
metrics.san.fc_target.show port=N1FC1
```

Output

---	OPERATIONS	THROUGHPUT	LATENCY	ERRORS
READ	7,121	251 MB/sec	3663.017ms	0
WRITE	3,927	126 MB/sec	2803.351ms	0
XCOPY	0	-	0.0ms	0
WRITE_SAME	0	-	0.0ms	0

metrics.san.fc_target.top ALL ROLES

CHANGE

Description Show top metrics for fc targets in system.

- The command output is filterable by:
  - Pool
  - Volume
  - Host
  - Initiator
- The number of metered entities can be determined
- The results can be sorted

Syntax

```
metrics.san.fc_target.top [pool=POOL] [vol={VOL|SNAP}] [host=HOST] [sort_by=TYPE] [results=Metrics Results]
```

Arguments

results	The number of results that will be displayed. Max value is 100
sort_by	operations, latency or throughput
host	Name of an existing host
vol	Name of an existing volume or snapshot
pool	Name of an existing pool

Example

```
metrics.san.fc_target.top
```

Output

TARGET	OPERATIONS	THROUGHPUT	LATENCY	ERRORS
57:42:b0:f0:00:0b:bb:31	5,680	106 MB/sec	7.91 ms	0
57:42:b0:f0:00:0b:bb:21	4,790	313 MB/sec	8.12 ms	0

metrics.san.host.show ALL ROLES

CHANGE

Description Show metrics for specific host in system.

Syntax

```
metrics.san.host.show host=HOST [interval=Metrics Interval] [count=Metrics Count]
```

Arguments

interval	Metrics refresh interval. Max value is 60
host	Name of an existing host
count	The number of times command will be executed

Example

```
metrics.san.host.show host=h1
```

Output

---	OPERATIONS	THROUGHPUT	LATENCY	ERRORS
READ	7,121	251 MB/sec	3663.017ms	0
WRITE	3,927	126 MB/sec	2803.351ms	0
XCOPY	0	-	0.0ms	0
WRITE_SAME	0	-	0.0ms	0

metrics.san.host.top ALL ROLES

CHANGE

Description Show top metrics for hosts in system.

Syntax

```
metrics.san.host.top [pool=POOL] [sort_by=TYPE] [results=Metrics Results]
```

Arguments

results	The number of results that will be displayed. Max value is 100
sort_by	operations, latency or throughput
pool	Name of an existing pool

Example

```
metrics.san.host.top
```

Output

HOST NAME	OPERATIONS	THROUGHPUT	LATENCY	ERRORS
10:00:00:00:c9:ac:c6:f3	5,680	106 MB/sec	7.91 ms	0
10:00:00:00:c9:f2:1d:d9	4,790	313 MB/sec	8.12 ms	0

metrics.san.iscsi.show ALL ROLES

CHANGE

Description Show metrics for all of the iSCSI in the system.

Syntax

```
metrics.san.iscsi.show [pool=POOL] [vol={VOL|SNAP}] [interval=Metrics Interval] [count=Metrics Count]
```

Arguments

count	The number of times command will be executed
interval	Metrics refresh interval. Max value is 60
vol	Name of an existing volume or snapshot
pool	Name of an existing pool

Example

```
metrics.san.iscsi.show
```

Output

---	OPERATIONS	THROUGHPUT	LATENCY	ERRORS
READ	7,121	251 MB/sec	3.017ms	0
WRITE	3,927	126 MB/sec	3.351ms	0
XCOPY	0	-	0.0ms	0
WRITE_SAME	0	-	0.0ms	0

metrics.san.iscsi_initiator.show ALL ROLES

CHANGE

Description Show metrics for specific iSCSI initiator in system.

Syntax

```
metrics.san.iscsi_initiator.show initiator=USEDISCSILOGGEDININITIATOR [interval=Metrics Interval] [count=Metrics Count]
```

Arguments

interval	Metrics refresh interval. Max value is 60
initiator	Logged in initiator
count	The number of times command will be executed

Example

```
metrics.san.iscsi_initiator.show initiator=172.16.35.74
```

Output

---	OPERATIONS	THROUGHPUT	LATENCY	ERRORS
READ	7,121	251 MB/sec	3663.017ms	0
WRITE	3,927	126 MB/sec	2803.351ms	0
XCOPY	0	-	0.0ms	0
WRITE_SAME	0	-	0.0ms	0

metrics.san.iscsi_initiator.top ALL ROLES

CHANGE

Description Show top metrics for iSCSI initiators in system.

Syntax

```
metrics.san.iscsi_initiator.top [pool=POOL] [host=HOST] [sort_by=TYPE] [results=Metrics Results]
```

Arguments

results	The number of results that will be displayed. Max value is 100
sort_by	operations, latency or throughput
host	Name of an existing host
pool	Name of an existing pool

Example

```
metrics.san.iscsi_initiator.top
```

Output

INITIATOR NAME	INITIATOR	OPERATIONS	THROUGHPUT	LATENCY	ERRORS
---	---	7,121	251 MB/sec	3.66 ms	0
---	---	3,927	126 MB/sec	2.80 ms	0

metrics.san.iscsi_initiator_ip.show ALL ROLES

CHANGE

Description Show metrics for specific iscsi initiator IP in system.

Syntax

```
metrics.san.iscsi_initiator_ip.show ip_address=IPADDRESS [interval=Metrics Interval] [count=Metrics Count]
```

Arguments

ip_address	A valid IPv4 address
count	The number of times command will be executed
interval	Metrics refresh interval. Max value is 60

Example

```
metrics.san.iscsi_initiator_ip.show ip_address=172.16.35.74
```

Output

---	OPERATIONS	THROUGHPUT	LATENCY	ERRORS
READ	7,121	251 MB/sec	3663.017ms	0
WRITE	3,927	126 MB/sec	2803.351ms	0
XCOPY	0	-	0.0ms	0
WRITE_SAME	0	-	0.0ms	0

metrics.san.iscsi_initiator_ip.top ALL ROLES

CHANGE

Description Show top metrics for iscsi initiator IPs in system.

Syntax

```
metrics.san.iscsi_initiator_ip.top [pool=POOL] [vol={VOL|SNAP}] [host=HOST] [sort_by=TYPE]
[results=Metrics Results]
```

Arguments

vol	Name of an existing volume or snapshot
pool	Name of an existing pool
results	The number of results that will be displayed. Max value is 100
sort_by	operations, latency or throughput
host	Name of an existing host

Example

```
metrics.san.iscsi_initiator_ip.top
```

Output

---	OPERATIONS	THROUGHPUT	LATENCY	ERRORS
READ	7,121	251 MB/sec	3663.017ms	0
WRITE	3,927	126 MB/sec	2803.351ms	0
XCOPY	0	-	0.0ms	0
WRITE_SAME	0	-	0.0ms	0

metrics.san.iscsi_target_ip.show ALL ROLES

CHANGE

Description Show metrics for specific iscsi target IP in system.

Syntax

```
metrics.san.iscsi_target_ip.show ip_address=IPADDRESS [interval=Metrics Interval] [count=Metrics
Count]
```

Arguments

count	The number of times command will be executed
interval	Metrics refresh interval. Max value is 60
ip_address	A valid IPv4 address

Example

```
metrics.san.iscsi_initiator_ip.show ip_address=172.16.35.74
```

Output

---	OPERATIONS	THROUGHPUT	LATENCY	ERRORS
READ	7,121	251 MB/sec	3663.017ms	0
WRITE	3,927	126 MB/sec	2803.351ms	0
XCOPY	0	-	0.0ms	0
WRITE_SAME	0	-	0.0ms	0

metrics.san.iscsi_target_ip.top

ALL ROLES

CHANGE

Description Show top metrics for iscsi target IPs in system.

Syntax

```
metrics.san.iscsi_target_ip.top [pool=POOL] [vol={VOL|SNAP}] [host=HOST] [sort_by=TYPE]
[results=Metrics Results]
```

Arguments

results	The number of results that will be displayed. Max value is 100
sort_by	operations, latency or throughput
host	Name of an existing host
vol	Name of an existing volume or snapshot
pool	Name of an existing pool

Example

```
metrics.san.iscsi_target_ip.top
```

Output

INITIATOR NAME	INITIATOR	OPERATIONS	THROUGHPUT	LATENCY	ERRORS
---	---	7,121	251 MB/sec	3.66 ms	0
---	---	3,927	126 MB/sec	2.80 ms	0

metrics.san.pool.show

ALL ROLES

CHANGE

Description Show metrics for specific pool in system.

Syntax

```
metrics.san.pool.show pool=POOL [interval=Metrics Interval] [count=Metrics Count]
```

Arguments

count	The number of times command will be executed
interval	Metrics refresh interval. Max value is 60
pool	Name of an existing pool

Example

```
metrics.san.pool.show pool=p1
```

Output

---	OPERATIONS	THROUGHPUT	LATENCY	ERRORS
READ	7,121	251 MB/sec	3663.017ms	0
WRITE	3,927	126 MB/sec	2803.351ms	0
XCOPY	0	-	0.0ms	0
WRITE_SAME	0	-	0.0ms	0

metrics.san.pool.top ALL ROLES

CHANGE

Description Show top metrics for pools in system.

Syntax

```
metrics.san.pool.top [sort_by=TYPE] [results=Metrics Results]
```

Arguments

results	The number of results that will be displayed. Max value is 100
sort_by	operations, latency or throughput

Example

```
metrics.san.pool.top
```

Output

POOL NAME	OPERATIONS	THROUGHPUT	LATENCY	ERRORS
p1	5,680	106 MB/sec	7.91 ms	0
p2	4,790	313 MB/sec	8.12 ms	0

metrics.san.system.show ALL ROLES

CHANGE

Description Show metrics for system.

Syntax

```
metrics.san.system.show [interval=Metrics Interval] [count=Metrics Count]
```

Arguments

interval	Metrics refresh interval. Max value is 60
count	The number of times command will be executed

Example

```
metrics.san.system.show
```

Output

---	OPERATIONS	THROUGHPUT	LATENCY	ERRORS
READ	7,121	251 MB/sec	3663.017ms	0
WRITE	3,927	126 MB/sec	2803.351ms	0
XCOPY	0	-	0.0ms	0
WRITE_SAME	0	-	0.0ms	0

metrics.san.vol.show ALL ROLES

CHANGE

Description Show metrics for specific volume in system.

Syntax

```
metrics.san.vol.show vol={VOL|SNAP} [interval=Metrics Interval] [count=Metrics Count]
```

Arguments

count	The number of times command will be executed
interval	Metrics refresh interval. Max value is 60
vol	Name of an existing volume or snapshot

Example

```
metrics.san.vol.show
```

Output

---	OPERATIONS	THROUGHPUT	LATENCY	ERRORS
READ	7,121	251 MB/sec	3663.017ms	0
WRITE	3,927	126 MB/sec	2803.351ms	0
XCOPY	0	-	0.0ms	0
WRITE_SAME	0	-	0.0ms	0

metrics.san.vol.top ALL ROLES

CHANGE

Description Show top metrics for volumes in system.

Syntax

```
metrics.san.vol.top [pool=POOL] [sort_by=TYPE] [results=Metrics Results]
```

Arguments

pool	Name of an existing pool
results	The number of results that will be displayed. Max value is 100
sort_by	operations, latency or throughput

Example

```
metrics.san.vol.top
```

Output

VOLUME NAME	OPERATIONS	THROUGHPUT	LATENCY	ERRORS
v1	5,680	106 MB/sec	7.91 ms	0
v2	4,790	313 MB/sec	8.12 ms	0

Chapter 43. HTTPS/SSL management

config.system.set_ssl_redirection ADMIN

CHANGE

Description Enables / disables the redirection from HTTP (80)to HTTPS (443).

Syntax

```
config.system.set_ssl_redirection redirect=YESNO
```

Arguments

redirect	Either yes, or no

Redirecting all sessions to use SSL

Example

```
config.system.set_ssl_redirection redirect=yes
```

Output

```
HTTPS redirection set to "enabled"
```

config.system.ssl_certificate.clear ADMIN

CHANGE

Description Clears existing SSL certificate, replaces it with a self-signed certificate.

Syntax

```
config.system.ssl_certificate.clear
```

Example

```
config.system.ssl_certificate.clear
```

Output

```
SSL certificate cleared, generated a self-signed certificate.
```

config.system.ssl_certificate.create_csr ADMIN

CHANGE

Description Create Certificate Sign Request (CSR) file. This file includes only the public key and is a more secure method of generating a certificate.

Syntax

```
config.system.ssl_certificate.create_csr [filepath=STR]
```

Arguments

filepath	Any sequence of characters

Create a certificate file and store it in the default location

Example

```
config.system.ssl_certificate.create_csr
```

Output

```
CSR saved to 'c:\temp\infinibox_csr.pem'
```

Create a certificate file and store it in a location other than the default

Example

```
config.system.ssl_certificate.create_csr filepath=c:\myCSR.pem
```

Output

```
CSR saved to 'c:\myCSR.pem'
```

config.system.ssl_certificate.query ADMIN

CHANGE

Description Show current certificate data.

Syntax

```
config.system.ssl_certificate.query
```

Example

```
config.system.ssl_certificate.query
```

Output

Public Key RAW	-----BEGIN PUBLIC KEY-----
	MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAjN + sR54V
	09 LtVvEEw7iPmYwqZ9idoL / JZu2Iabd6sNr / vGGQEQeORcUjP8wB
	qeiS3C2JiOsVapd6idXqpEsS4CUn / yv4Wo6Oe3n2ikzIacYzfSnA
	vSwoHt896YCrqR / qr3oZjigergBFMqXu + BsXPEwT6BnPvXmLI0xd
	wt8xQ3ghilsDx0euCdn4Oaf9dySfYYmNAPo8GEdklBLuYLQV / Tqf
	XxxSAjBGQNC2Ye / mXhYbtvLUjAyWnxP47vpPlRt + GXOsHgh1tbcU

	vQIDAQAB
	-- -- - END PUBLIC KEY-- -- -
Public Key Modulus	8CDFAC479E15D465747952528C5CB2D3D2ED56F104C3B88F998C
	5E AA44B12E02527FF2BF85A8E8E7B79F68A4CC869C6337D29C0753195AD176D6998E2BD2C281EDF3DE980ABA9
	9 D9F839A7FD77249F61898D00FA3C1847649412EE60B415FD3A9FBADDD69914B8E2E0905F1C5202304640D0B6
Public Key Algorithm	RSA
Certificate Issued On	2016-11-08 10:00:00
Certificate Raw	-----BEGIN CERTIFICATE-----
	MIIDBzCCAfGgAwIBAgIGAVhEHeLHMAsGCSqGSIb3DQEBCzA7MQsw
	UzESMBAGA1UEAwwJSW5maW5pbGFiMRgwFgYDVQQKDA9JbmZpbmlk
	HhcNMTYxMTA4MTMyNTU5WhcNMjExMTA4MTMyNTU5WjBSMQswCQYD
	MCcGA1UEAwwgYm94LXJleWZtYW4ubGFiLmlsLmluZmluaWRhdC5j
	BAoMD0luZmluaWRhdCwgTHRkLjCCASIwDQYJKoZIhvcNAQEBBQAD
	ggEBAIzfrEeeFdRldHlSUoxcstPS7VbxBMO4j5mMKmfYnaC / yWbt
	kBEHjkXFIz / MAb + 8 rn9sEpfNH6noktwtiYjrFWqXeonV6qRLEuAl
	9 opMyGnGM30pwHUxla0XbWmY4r0sKB7fPemAq6kf6q96GY4oHq4A
	E + gZz715iyNMXdVewUUlL0D0VMLfMUN4IYpbA8dHrgnZ + Dmn / Xck
	ZJQS7mC0Ff06n7rd1pkUuOLgkF8cUgIwRkDQtmHv5l4WG7by1IwM
	fhlzrB4IdbW3FD2GKyce4uv5870CAwEAATALBgkqhkiG9w0BAQsD
	qh9nUnkR1fPW8YykChSoQIYh63c93ON2xPhNeCd6fkEZcYf4w18t
	99E3 y + RIpM4hNenOmCeWcWoUacEI4YIZQqhk2QVHTbyA / tTGEh9m
	uxjm9Fu22WLmqXb8Ce / TUwtZUJWP6T95WGNYVvMdw7s0f0Uw0JOW
	gyk54Pg477LHjjfH35ssC7cI1V6zQBiSzOLDR3BPFklp / +vREd0x
	tIuU5FA + NIGUIS9kkOoR / 3 SVr99Nfaowjbwv0e / +S7uRfcYYLoTp
	z5mr / 2 yGTun0wo4 =
	-- -- - END CERTIFICATE-- -- -
Certificate Version	3
Certificate Signature Algorithm	SHA256withRSA
Certificate Serial Number	158441DE2C7
Certificate Expired	no
Certificate Expires On	2021-11-08 15:00:00
Certificate Issued By Country	US
Certificate Issued By Organization	Infinidat, Ltd.
Certificate Issued By Cn	Infinilab

Certificate Issued To Country	US
Certificate Issued To Organization	Infinidat, Ltd.
Certificate Issued To Cn	ibox9999.lab.il.infinidat.com

config.system.ssl_certificate.upload ADMIN CHANGE

Description Sets the SSL certificate for the system.

InfiniBox validates that:

- The certificate file is in PEM format
- The certificate digital signature matches the certificate file content (the certificate is valid)

Syntax

```
config.system.ssl_certificate.upload filename=STR
```

Arguments

filename	Any sequence of characters

Example

```
config.system.ssl_certificate.set_ssl filename=c:\cert.pem
```

Output

```
System SSL Certificate uploaded
```

config.system.ssl_certificate.upload_signed_csr ADMIN CHANGE

Description Sets the certificate using a signed CSR file.

Syntax

```
config.system.ssl_certificate.upload_signed_csr filename=STR
```

Arguments

filename	Any sequence of characters

Example

```
config.system.ssl_certificate.upload_signed_csr
```

Output

```
Signed CSR uploaded successfully
```

Chapter 44. QoS

config.qos_policy.assign ADMIN CHANGE

Description Assign entity to policy.

Each policy is effective to only one type of entity:

- Volume
- Filesystem
- Pool-volume - the policy impacts the volumes that belong to this pool
- Pool-filesystem - the policy impacts the filesystems that belong to this pool

Syntax

```
config.qos_policy.assign policy=QOSPOLICY [vol={VOL|SNAP}[,{VOL|SNAP},...]] [pool=POOL[,POOL,...]]
```

Arguments

pool	Name of an existing pool (multiple values, separated by commas)
vol	Name of an existing volume or snapshot (multiple values, separated by commas)
policy	QoS policy

Assigning a volume to a policy

Example

```
qos_policy.assign policy=q1 vol=v1
```

Output

```
VOLUME 'v1' was assigned to policy 'q1'
```

Assigning a filesystem to a policy

Example

```
qos_policy.assign policy=q2 fs=fs1
```

Output

```
FILESYSTEM 'fs1' was assigned to policy 'q2'
```

config.qos_policy.assignment_query ALL ROLES

CHANGE

Description List entities that are associated with a policy.

Syntax

```
config.qos_policy.assignment_query [policy=QOSPOLICY] [type=QoS policy type]
```

Arguments

policy	QoS policy
type	VOLUME or POOL_VOLUME

Example

```
qos_policy.assignment_query
```

Output

NAME	TYPE	POLICY	ENTITIES
v1	VOLUME	q1	-
fs1	FILESYSTEM	q2	-
p1	POOL_VOLUME	q4	1
p1	POOL_FILESYSTEM	q3	1

config.qos_policy.create ADMIN

CHANGE

Description Create a new QoS policy.

Syntax

```
config.qos_policy.create name=NAME type=QoS policy type [max_ops=QOSOPSORNONE]
[max_throughput=MAXMIBPSORNONE] [burst=BURST] [burst_factor=BURSTFACTOR]
```

Arguments

`name`	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
`burst_factor`	QoS policy burst factor (Min value: None, Max value: None)
`burst`	Enable or disable Burst for QoS policy
`max_throughput`	NONE, or qoS policy max throughput in MiB/s
`max_ops`	NONE, or qoS policy max operations per second
`type`	VOLUME or POOL_VOLUME

Creating a policy with max IOPS, and a burst

Example

```
qos_policy.create name=q6 type=volume max_ops=2000 burst_factor=4 burst=yes
```

Output

```
QoS policy "q6" created
```

Creating a policy with max IOPS and no burst

Example

```
qos_policy.create name=q5 type=volume max_ops=2000
```

Output

```
QoS policy "q5" created
```

config.qos_policy.delete ADMIN

CHANGE

Description Delete a policy

- There is no need to unassigned entities from a policy prior to the deletion
- Entities that were assigned to this policy are no longer assigned to a policy

Syntax

```
config.qos_policy.delete policy=QOSPOLICY
```

Arguments

`policy`	QoS policy

Example

```
qos_policy.delete policy=q1
```

Output

```
QoS policy "q1" deleted
```

config.qos_policy.modify ADMIN

CHANGE

Description Modify a policy

- Rename the policy
- Change the max operations or max throughput settings
- Add a burst
- Activate/deactivate the burst

Syntax

```
config.qos_policy.modify policy=QOSPOLICY [new_name=NAME] [max_ops=QOSOPSORNONE]
[max_throughput=MAXMIBPSORNONE] [burst=BURST] [burst_factor=BURSTFACTOR]
```

Arguments

`burst_factor`	QoS policy burst factor (Min value: None, Max value: None)
`burst`	Enable or disable Burst for QoS policy
`max_throughput`	NONE, or qoS policy max throughput in MiB/s
`max_ops`	NONE, or qoS policy max operations per second
`new_name`	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@()[]$=!-#{}%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
`policy`	QoS policy

Deactivating the burst

Example

```
qos_policy.modify policy=policy-1 burst=no
```

Output

```
QoS policy "policy-1" updated
```

Changing the burst factor

Example

```
qos_policy.modify policy=policy-1 burst_factor=4 burst=yes
```

Output

```
QoS policy "policy-1" updated
```

Renaming a policy

Example

```
qos_policy.modify policy=q1 new_name=q2
```

Output

```
Policy "q1" renamed to "q2"
```

Changing the max operations and max throughput parameters

Example

```
qos_policy.modify policy=q1 max_ops=3000
```

Output

```
QoS policy "q1" updated
```

config.qos_policy.query ALL ROLES

CHANGE

Description List existing policies.

Syntax

```
config.qos_policy.query [policy=QOSPOLICY] [type=QoS policy type]
```

Arguments

policy	QoS policy
type	VOLUME or POOL_VOLUME

Example

```
qos_policy.query
```

Output

NAME	TYPE	MAX OPS	MAX THROUGHPUT	BURST FACTOR
q1	VOLUME	3000	UNLIMITED	4
q2	FILESYSTEM	3000	UNLIMITED	-
q3	POOL_VOLUME	10000	UNLIMITED	-
q4	POOL_FILESYSTEM	10000	UNLIMITED	-

config.qos_policy.unassign ADMIN

CHANGE

Description **Remove entity from policy.**

Unassigning an entity from a policy is done in either of the following ways:

- Specifying the entity
- Specifying the policy and then the entity

It is impossible to unassign all of the entities of a policy.

Syntax

```
config.qos_policy.unassign [vol={VOL|SNAP}[,{VOL|SNAP},...]] [pool=POOL[,POOL,...]]
[policy=QOSPOLICY]
```

Arguments

policy	QoS policy
pool	Name of an existing pool (multiple values, separated by commas)
vol	Name of an existing volume or snapshot (multiple values, separated by commas)

Specifying the policy and the entity.

Example

```
qos_policy.unassign policy=q1 vol=v1
```

Output

```
Entity was unassigned from QoS policy
```

Specifying the entity.

Example

```
qos_policy.unassign vol=v1
```

Output

```
Entity was unassigned from QoS policy
```

Special Commands

%clear ALL ROLES

CHANGE

Description Clear screen.

Syntax

```
%clear
```

Example

```
%clear
```

Output

```
The screen is cleared and only the command prompt is displayed.
```

%count ALL ROLES

CHANGE

Description **Toggle count output.**

This toggle changes the output of any query command to display the number of the queried entities (rather than a list of these entities). This command can be used in two ways:

- Run `%count` and then any query command
  - Run `%count` again to toggle-off the count option
- Run any query command with: `--format=count`

Syntax

```
%count
```

Toggling-on the count output

Example

```
%count
```

Output

```
Count output enabled
Use %count again to disable count output
```

Toggling-off the count output

Example

```
%count
```

Output

```
Count output disabled
Use %count again to enable count output
```

Running a query command when the count toggle in on

- In this example, there are 10 volumes on the system

Example

```
vol.query
```

Output

```
10
```

Running a query command regardless of the state of the count toggle

- In this example, there are 10 volumes on the system

Example

```
vol.query --format=count
```

Output

```
10
```

%csv ALL ROLES

CHANGE

Description Toggle the output to CSV mode.

Enter %csv and then enter a query command. The output will be displayed in CSV format. Enter %csv again to resume to a tabular output format.

Syntax

```
%csv
```

Example

```
%csv
```

Output

```
CSV output enabled
```

%debug ALL ROLES

CHANGE

Description Toggle debug prints.

Enter %debug to toggle to debug mode. Enter a command. The command output will be preceded by debugging messages. Enter %debug again to exit from the Debug mode.

Syntax

```
%debug
```

Example

```
%debug
```

Output

```
Debug: True
```

%json ALL ROLES

CHANGE

Description Toggle JSON output

When toggled, the command output will be displayed in JSON format.

Syntax

```
%json
```

Example

```
%json
```

Output

```
{
    "command": "%json",
    "result": "JSON output enabled\n\nUse %json again to disable json output"
}
```

%logstart ALL ROLES

CHANGE

Description Start command logger.

Enter %logstart to start logging the session you are currently in. Providing a file name is mandatory (unless already provided in the configuration file). You may provide values for other parameters (none of them is mandatory). The name parameter saves the log to a file with the specified name and sets the mode to append. The mode parameter determines the way the log is saved into a file. Append mode keeps logging to an existing log file. Override mode overrides an existing log file. The output parameter logs the output of each entered command. The timestamp parameter adds a time stamp to each of the logged commands.

Syntax

```
%logstart [file=STR] [mode=LOGGINGMODE] [output=BOOL] [timestamps=BOOL]
```

Arguments

`timestamps`	A boolean: true, false, yes, no, on, off
`output`	A boolean: true, false, yes, no, on, off
`mode`	overwrite or append
`file`	Any sequence of characters

Example

```
%logstart file=/Users/user-name/tmp/log.txt mode=append output=yes timestamps=yes
```

Output

```
Logger activated
```

%logstate ALL ROLES

CHANGE

Description Display command logger state.

This command displays the state of the logging (either on or off).

Syntax

```
%logstate
```

Example

```
%logstate
```

Output

```
Logging is on
```

%logstop ALL ROLES

CHANGE

Description Stop command logger.

This command stops logging.

Syntax

```
%logstop
```

Example

```
%logstop
```

Output

```
Logger deactivated
```

%raw ALL ROLES

CHANGE

Description Toggle raw API output.

Enables displaying the JSON query of the commands output.

Syntax

```
%raw
```

Example

```
%raw
```

Output

```
Command prompt.
```

%user ALL ROLES CHANGE

Description Displays the name of the active user.

Syntax

```
%user
```

Example

```
%user
```

Output

```
Admin
```

%version ALL ROLES CHANGE

Description Displays the version of the InfiniShell client.

Syntax

```
%version
```

Example

```
%version
```

Output

```
1.5.0.x
```

NEW COMMANDS

config.net_space.set_replication_type ADMIN POOL ADMIN CHANGE

Description Sets a bandwidth limitation on a Network Space per node. This limit is configured on the Interface used by the Network space to allow traffic shaping and avoid bandwidth starvation.

Syntax

```
config.net_space.set_replication_type net_space=REPLICATIONNETWORKSPACE
replication_type=NETWORKRMRTYPE
```

Arguments

`net_space`	Name of an existing replication network space
`replication_type`	Network space replication type

config.system.set_fqdn ADMIN CHANGE

Description Enables / disables default pool ssd cache.

Syntax

```
config.system.set_fqdn fqdn=STR
```

Arguments

fqdn	Any sequence of characters

fs.treeq.create ADMIN

CHANGE

Description Create new filesystem TreeQ.

Syntax

```
fs.treeq.create fs={FS|SNAP} name=TREEQNAME [path=TREEQPATH] [mode=TREEQMODE]
[inode_quota=TREEQINODEQUOTA] [soft_inode_quota=TREEQSOFTINODEQUOTA]
[capacity_quota=TREEQCAPACITYQUOTA] [soft_capacity_quota=TREEQSOFTCAPACITYQUOTA]
```

Arguments

fs	Name of an existing filesystem or snapshot
name	TreeQ name
path	TreeQ path
mode	TreeQ mode
inode_quota	A positive number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1M, 1000000
soft_inode_quota	An integer number equal to or lesser than 2
capacity_quota	A positive number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1TB, 1000000000000
soft_capacity_quota	A positive number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1TB, 1000000000000

fs.treeq.delete ADMIN

CHANGE

Description Delete filesystem TreeQ.

Syntax

```
fs.treeq.delete fs={FS|SNAP} treeq=TREEQNAME
```

Arguments

fs	Name of an existing filesystem or snapshot
treeq	TreeQ name

fs.treeq.modify ADMIN

CHANGE

Description Modify filesystem TreeQ.

Syntax

```
fs.treeq.modify fs={FS|SNAP} [treeq=TREEQNAME] [path=TREEQPATH] [inode_quota=TREEQINODEQUOTAORNONE]
[soft_inode_quota=TREEQSOFTINODEQUOTAORNONE] [capacity_quota=TREEQCAPACITYQUOTAORNONE]
[soft_capacity_quota=TREEQSOFTCAPACITYQUOTAORNONE]
```

Arguments

`treeq`	TreeQ name
`fs`	Name of an existing filesystem or snapshot
`path`	TreeQ path
`inode_quota`	A positive number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1M, 1000000
`soft_inode_quota`	An integer number equal to or lesser than 2
`capacity_quota`	A positive number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1TB, 1000000000000
`soft_capacity_quota`	A positive number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1TB, 1000000000000

fs.treeq.query ADMIN

CHANGE

Description List existing treeqs.

Syntax

```
fs.treeq.query fs={FS|SNAP} [treeq=TREEQNAME] [path=TREEQPATH] [inodes_state=TREEQSTATE]
[capacity_state=TREEQSTATE]
```

Arguments

`capacity_state`	TreeQ state
`inodes_state`	TreeQ state
`path`	TreeQ path
`treeq`	TreeQ name
`fs`	Name of an existing filesystem or snapshot

fs.treeq.rename ADMIN

CHANGE

Description Rename filesystem TreeQ.

Syntax

```
fs.treeq.rename fs={FS|SNAP} treeq=TREEQNAME new_name=TREEQNAME
```

Arguments

`new_name`	TreeQ name
`treeq`	TreeQ name
`fs`	Name of an existing filesystem or snapshot

REMOVED COMMANDS

cluster.clear_chap ALL ROLES

CHANGE

Description Clears CHAP settings.

- This command clears the CHAP username and secret from both initiator and target
- This command clears the CHAP authentication type and sets the cluster CHAP to NONE

Syntax

```
cluster.clear_chap cluster=CLUSTER
```

Arguments

`cluster`	Name of an existing cluster

Example

```
cluster.clear_chap cluster=c1
```

Output

```
Cluster "c1" CHAP settings cleared
```

cluster.set_auth_type ADMIN POOL ADMIN CHANGE

Description Set iSCSI authentication method for all hosts in cluster.

- See host.set_auth_type for details
- Use cluster.add_host to add hosts to the cluster
- Use cluster.host_query to see which hosts belong to the cluster

Syntax

```
cluster.set_auth_type cluster=CLUSTER type=METHOD
```

Arguments

`type`	Authentication method required of host when connecting through iSCSI: NONE, CHAP or MUTUAL_CHAP
`cluster`	Name of an existing cluster

Setting the authentication type to CHAP for host that belong to the same cluster

Example

```
cluster.set_auth_type name=c1 type=CHAP
```

Output

```
Host "h1": iSCSI authentication type set to CHAP
Host "h2": iSCSI authentication type set to CHAP
```

cluster.set_chap ADMIN POOL ADMIN CHANGE

Description Set iSCSI CHAP usernames and secrets for all hosts in cluster.

Syntax

```
cluster.set_chap cluster=CLUSTER [inbound_user=STR] [inbound_secret=STR] [outbound_user=STR]
[outbound_secret=STR]
```

Arguments

`outbound_secret`	Any sequence of characters
`outbound_user`	Any sequence of characters
`inbound_secret`	Any sequence of characters
`inbound_user`	Any sequence of characters
`cluster`	Name of an existing cluster

Setting the inbound name and secret

Example

```
cluster.set_chap name=c1 inbound_user=admin inbound_secret=cbA10V2yk6mhLEyV3PAe
```

Output

```
Cluster "c1" CHAP settings updated
```

config.cache.ssd_default ADMIN

CHANGE

Description Set SSD cache for new pools.

Run the command again to toggle the default for new pools to be SSD disabled.

Syntax

```
config.cache.ssd_default ssd=YESNO
```

Arguments

ssd	Either yes, or no

Example

```
config.cache.ssd_default ssd=yes
```

Output

```
SSD cache default for new pools set to "enable"
```

config.ethernet.interface.set_rate_limit

CHANGE

Description Allocates bandwidth to a specific Interface.

Set a rate limit in Mbps units to a specific interface. To remove the limit, run the command again with the value NONE.

Syntax

```
config.ethernet.interface.set_rate_limit
```

Example

```
config.ethernet.interface.set_rate_limit name=interface1 limit=500
```

Output

```
Network interface interface1 rate limit set to 500Mbps
```

Example

```
config.ethernet.interface.set_rate_limit name=interface1 limit=NONE
```

Output

```
Network interface interface1 rate limit removed
```

config.fc.soft_target.query ALL ROLES

CHANGE

Description List FC soft targets.

Syntax

```
config.fc.soft_target.query [target=WWPN[,WWPN,...]] [switch=SWITCH] [node=NODE] [home=YESNO]
```

Arguments

home	Either yes, or no
node	Node name
switch	Name of an existing fc switch
target	Name of an existing fc soft target (multiple values, separated by commas)

Example

```
config.fc.soft_target.query
```

Output

WWPN	Node	Port	Switch	Is on home node
20:01:74:2B:0F:00:4E:21	1	1	brocade01	Yes
20:01:74:2B:0F:00:4E:21	2	1	brocade02	Yes
20:01:74:2B:0F:00:4E:21	3	1	brocade03	Yes
20:01:74:2B:0F:00:4E:21	2	5	cisco01	No
20:01:74:2B:0F:00:4E:21	2	5	cisco03	Yes
20:01:74:2B:0F:00:4E:21	3	5	cisco01	Yes

config.net_space.disable ADMIN CHANGE

Description Disabled a Network Space, deactivates its service and IP addresses.

Syntax

```
config.net_space.disable name=NETWORKSPACE
```

Arguments

name	Name of an existing network space

Example

```
config.net_space.disable name=iscsi-data1
```

Output

```
Network space "iscsi-data1" disabled
```

config.net_space.enable ADMIN CHANGE

Description Enables a Network Space, activates its service and IP addresses.

Syntax

```
config.net_space.enable name=NETWORKSPACE
```

Arguments

name	Name of an existing network space

Example

```
config.net_space.enable name=iscsi-data1
```

Output

```
Network space "iscsi-data1" enabled
```

config.net_space.ip.set_management_ip ADMIN CHANGE

Description Change management IP for the network space.

Syntax

```
config.net_space.ip.set_management_ip net_space=NETWORKSPACE ip=IPADDRESS
```

Arguments

ip	A valid IPv4 address
net_space	Name of an existing network space

config.net_space.set_service ADMIN

CHANGE

Description Set a service for a network space.

Syntax

```
config.net_space.set_service name=NETWORKSPACE service=SERVICE
```

Arguments

service	NAS, ISCSI or REPLICATION
name	Name of an existing network space

Example

```
config.net_space.set_service name=default_nas_space service=NAS_SERVICE
```

Output

```
Network space default_nas_space service set to NAS_SERVICE
```

config.system.set_default_gateway

CHANGE

Description Set management interface default gateway IP.

Syntax

```
config.system.set_default_gateway
```

Example

```
config.system.set_default_gateway ip=172.16.95.255
```

Output

```
Management interface default gateway IP set to 172.16.95.255
```

config.system.set_management_ip

CHANGE

Description Set management interface IP.

Syntax

```
config.system.set_management_ip
```

Example

```
config.system.set_management_ip ip=172.16.65.75 netmask=255.255.224.0
```

Output

```
Management interface IP set to 172.16.65.75 / 255.255.224.0
```

drive.phase_in

CHANGE

Description Initiate drive phase-in process.

This command operates on phased-out disks only. Test the drive prior to phasing it in.

Syntax

```
drive.phase_in
```

Example

```
drive.phase_in drive=E1D01
```

Output

```
Drive 1 of enclosure 1 is phasing-in
```

drive.phase_out

CHANGE

Description Initiate drive phase-out process.

Run this command prior to the drive replacement. This command operates on phased-in drives only.

Syntax

```
drive.phase_out
```

Example

```
drive.phase_out drive=E1D01
```

Output

```
Drive 1 of enclosure 1 is phasing-out
```

drive.test

CHANGE

Description Initiate drive testing process (required before phase-in is allowed).

Following the drive replacement, test the phased-out drive before you phase-it in.

Syntax

```
drive.test
```

Example

```
drive.test drive=E1D01
```

Output

```
Tested drive 1 of enclosure 1
```

fs.assign_to_qos_policy ADMIN

CHANGE

Description Assign filesystem to policy.

Syntax

```
fs.assign_to_qos_policy fs=FS policy=QOSPOLICY
```

Arguments

fs	Name of an existing filesystem
policy	QOS policy

fs.clone.cache ADMIN POOL ADMIN

CHANGE

Description Enable or disable SSD cache for a filesystem clone.

The clone cache settings are inherited from its ancestors.

Syntax

```
fs.clone.cache clone={FS|SNAP} ssd_cache=YESNO [recursive=YESNO]
```

Arguments

recursive	Either yes, or no
ssd_cache	Either yes, or no
clone	Name of an existing filesystem or snapshot

Example

```
fs.clone.cache name=fs1s1c1 ssd=yes recursive=yes
```

Output

```
Filesystem clone fs1s1c1 and all descendants SSD cache enabled
```

fs.clone.create

ADMIN POOL ADMIN

CHANGE

Description Create a new filesystem clone from filesystem snapshot.

Syntax

```
fs.clone.create snap={FS|SNAP} name=NAME [ssd_cache=YESNO]
```

Arguments

name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@() []$=!-#{}%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
snap	Name of an existing filesystem or snapshot
ssd_cache	Either yes, or no

Example

```
fs.clone.create snap=fs1c1 name=fs1s1c1
```

Output

```
Filesystem clone fs1s1c1 created
```

fs.clone.delete

ADMIN POOL ADMIN

CHANGE

Description Delete filesystem clone.

Syntax

```
fs.clone.delete clone={FS|SNAP}[,{FS|SNAP},...]
```

Arguments

clone	Name of an existing filesystem or snapshot (multiple values, separated by commas)

Example

```
fs.clone.delete name=fs1s1c1
```

Output

```
Filesystem clone fs1s1c1 deleted
```

fs.clone.export

CHANGE

Description Export filesystem clone.

Syntax

```
fs.clone.export
```

Example

```
fs.clone.export name=CLONE export=STR
```

Output

fs.clone.export_query

CHANGE

Description List existing filesystem clone exports.

Syntax

```
fs.clone.export_query
```

Example

```
fs.clone.export_query name=CLONE
```

Output

fs.clone.query ALL ROLES

CHANGE

Description List existing clones.

Syntax

```
fs.clone.query [clone={FS|SNAP}[,{FS|SNAP},...]] [source={FS|SNAP}] [pool=POOL]
[unit=NASCAPACITYUNIT] [has_exports=YESNO]
```

Arguments

has_exports	Either yes, or no
unit	B, G (or GB), GiB, T (or TB), TiB
pool	Name of an existing pool
source	Name of an existing filesystem or snapshot
clone	Name of an existing filesystem or snapshot (multiple values, separated by commas)

Example

```
fs.clone.query
```

Output

NAME	THIN	SIZE	USED	ALLOCATED	TREE ALLOCATED	POOL	WP	CREATED AT
fs1s1c1	no	1.00 GB	327.68 KB	1.00 GB	0	pool1	no	2015-06-01 10:00:00

fs.clone.rename ADMIN POOL ADMIN

CHANGE

Description Rename filesystem clone.

Syntax

```
fs.clone.rename clone={FS|SNAP} new_name=NAME
```

Arguments

new_name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
clone	Name of an existing filesystem or snapshot

Example

```
fs.clone.rename name=fs1s1c1 new_name=fs1s1clone1
```

Output

```
Filesystem clone fs1s1c1 renamed to fs1s1clone1
```

fs.clone.resize ADMIN POOL ADMIN

CHANGE

Description Resize filesystem clone.

Syntax

```
fs.clone.resize clone={FS|SNAP} size=SIZEDELTA
```

Arguments

size	A number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1TB, 1000000000000. When preceded by a plus or minus sign, represents a delta; e.g. 3000000m (absolute size), +2000GB (positive delta), -1.5T (negative delta)
clone	Name of an existing filesystem or snapshot

Example

```
fs.clone.resize name=fs1s1c1 size=10GB
```

Output

```
Filesystem clone fs1s1c1 resized
```

fs.clone.restore ADMIN POOL ADMIN

CHANGE

Description **Restore a filesystem clone from one of its snapshots.**

This command restores a clone. To restore a filesystem, use the fs.restore command. The clone can be restored from any of its snapshots or clones.

Syntax

```
fs.clone.restore clone={FS|SNAP} source=SNAP
```

Arguments

source	Name of an existing filesystem snapshot
clone	Name of an existing filesystem or snapshot

Example

```
fs.clone.restore name=fs1s1c1 snap=fs1s1c1s1
```

Output

```
Filesystem clone fs1s1c1 restored from snapshot fs1s1c1s1
```

fs.clone.tree ALL ROLES

CHANGE

Description **Display the hierarchy of clones and their descendants.**

To view the clone ancestors, use fs.snap.query, fs.query, fs.tree, or fs.snap.tree.

Syntax

```
fs.clone.tree clone={FS|SNAP} [unit=NASCAPACITYUNIT]
```

Arguments

unit	B, G (or GB), GiB, T (or TB), TiB
clone	Name of an existing filesystem or snapshot

Note that in the following example, the snapshot is a descendant of the clone.

Example

```
fs.clone.tree name=fs1s1c1
```

Output

NAME	TYPE	SIZE
fs1s1c1	CLONE	0.002 TB
|----fs1s1c1s1	SNAP	0.002 TB

fs.clone.unexport

CHANGE

Description Unexport filesystem clone.

Syntax

```
fs.clone.unexport
```

Example

```
fs.clone.unexport name=CLONE export=STR
```

Output

fs.clone.write_enable ADMIN POOL ADMIN CHANGE

Description Enable write access on filesystem clone.

Syntax

```
fs.clone.write_enable name={FS|SNAP} [recursive=YESNO]
```

Arguments

name	Name of an existing filesystem or snapshot
recursive	Either yes, or no

Example

```
fs.clone.write_enable name=fs1s1c1
```

Output

```
Filesystem clone fs1s1c1 write access enabled
```

fs.clone.write_protect ADMIN POOL ADMIN CHANGE

Description Disable write access on filesystem clone.

Syntax

```
fs.clone.write_protect clone={FS|SNAP} [recursive=YESNO]
```

Arguments

recursive	Either yes, or no
clone	Name of an existing filesystem or snapshot

Example

```
fs.clone.write_protect name=fs1s1c1
```

Output

```
Filesystem clone fs1s1c1 write access disabled
```

fs.export CHANGE

Description Export a filesystem.

Syntax

```
fs.export
```

Example

```
fs.export name=fs1 export_path=/export/fs1
```

Output

```
Export /export/fs1 created
```

fs.export.details CHANGE

Description List existing filesystem exports in detailed format.

Syntax

```
fs.export.details
```

Example

```
fs.export.details export_path=EXPORT
```

Output

```
Awaits CLI-1371
```

fs.export_query

CHANGE

Description List existing filesystem exports.

Syntax

```
fs.export_query
```

Example

```
fs.export_query name=fs1
```

Output

EXPORT PATH	INTERNAL PATH	FILESYSTEM	TYPE	ENABLED
/export/fs1	/	fs1	FILESYSTEM	yes

fs.remove_from_qos_policy ADMIN

CHANGE

Description Remove fs from policy.

Syntax

```
fs.remove_from_qos_policy fs=FS policy=QOSPOLICY
```

Arguments

fs	Name of an existing filesystem
policy	QOS policy

fs.snap.assign_to_qos_policy ADMIN

CHANGE

Description Assign filesystem to policy.

Syntax

```
fs.snap.assign_to_qos_policy snap=SNAP policy=QOSPOLICY
```

Arguments

snap	Name of an existing filesystem snapshot
policy	QOS policy

fs.snap.export

CHANGE

Description Export filesystem snapshot.

Syntax

```
fs.snap.export
```

Example

```
fs.snap.export name=SNAP export=STR
```

Output

fs.snap.export_query

CHANGE

Description List existing filesystem snapshot exports.

Syntax

```
fs.snap.export_query
```

Example

```
fs.snap.export_query name=SNAP
```

Output

fs.snap.remove_from_qos_policy ADMIN

CHANGE

Description Remove fs from policy.

Syntax

```
fs.snap.remove_from_qos_policy snap=SNAP policy=QOSPOLICY
```

Arguments

policy	QOS policy
snap	Name of an existing filesystem snapshot

fs.snap.unexport

CHANGE

Description Unexport filesystem snapshot.

Syntax

```
fs.snap.unexport
```

Example

```
fs.snap.unexport name=SNAP export=STR
```

Output

fs.unexport

CHANGE

Description Unexport filesystem.

Syntax

```
fs.unexport
```

Example

```
fs.unexport name=FS export=STR
```

Output

metrics.san.iscsi_target.show ALL ROLES

CHANGE

Description Show metrics for specific iscsi target in system.

Syntax

```
metrics.san.iscsi_target.show ip_address=IPADDRESS [interval=Metrics Interval] [count=POSITIVE]
```

Arguments

ip_address	A valid IPv4 address
interval	Metrics refresh interval. Max value is 60
count	A positive integer number, greater than zero

Example

```
metrics.san.iscsi_target.show ip_address=172.16.35.74
```

Output

---	OPERATIONS	THROUGHPUT	LATENCY	ERRORS
READ	7,121	251 MB/sec	3663.017ms	0
WRITE	3,927	126 MB/sec	2803.351ms	0
XCOPY	0	-	0.0ms	0
WRITE_SAME	0	-	0.0ms	0

metrics.san.iscsi_target.top ALL ROLES

CHANGE

Description Show top metrics for iscsi targets in system.

Syntax

```
metrics.san.iscsi_target.top [pool=POOL] [vol={VOL|SNAP}] [host=HOST]
[initiator=USEDLOGGEDININITIATOR] [sort_by=TYPE] [results=POSITIVE]
```

Arguments

pool	Name of an existing pool
results	A positive integer number, greater than zero
sort_by	ops, external_latency or throughput
initiator	A WWN or IQN address. A WWN address consists of a 16-digit hexadecimal number, optionally delimited by colons to 8 groups of two digits each. An IQN address begins with the string "iqn.", optionally followed by a domain registration date, followed by an organizational naming authority, and optionally followed by a colon and a custom string of choosing, e.g "iqn.2001-04.com.example:diskarrays-sn-a8675309".
host	Name of an existing host
vol	Name of an existing volume or snapshot

Example

```
metrics.san.iscsi_target.top
```

Output

TARGET NAME	TARGET	OPERATIONS	THROUGHPUT	LATENCY	ERRORS
---	---	7,121	251 MB/sec	3.66 ms	0
---	---	3,927	126 MB/sec	2.80 ms	0

pool.assign_to_policy ADMIN

CHANGE

Description Assign pool to policy.

Syntax

```
pool.assign_to_policy pool=POOL policy=QOSPOLICY
```

Arguments

pool	Name of an existing pool
policy	QOS policy

pool.cache ADMIN POOL ADMIN CHANGE

Description Enable or disable SSD cache.

Enter the command once in order to enable the pool to use SSD cache. Enter the command again to disable SSD cache.

Syntax

```
pool.cache pool=POOL ssd_cache=YESNO
```

Arguments

pool	Name of an existing pool
ssd_cache	Either yes, or no

Example

```
pool.cache name=pool1 ssd=yes
```

Output

```
Pool pool1 SSD cache enabled
```

pool.remove_from_policy ADMIN CHANGE

Description Remove pool from policy.

Syntax

```
pool.remove_from_policy pool=POOL policy=QOSPOLICY
```

Arguments

pool	Name of an existing pool
policy	QOS policy

pool.set_cache ADMIN POOL ADMIN CHANGE

Description Enable or disable SSD cache.

Syntax

```
pool.set_cache name=POOL ssd=YESNO
```

Arguments

ssd	Either yes, or no
name	Name of an existing pool

qos.policy.assign_dataset ADMIN CHANGE

Description Assign dataset to policy.

Syntax

```
qos.policy.assign_dataset policy=QOSPOLICY dataset={DATASET|SNAP}[,{DATASET|SNAP},...]
```

Arguments

dataset	Name of an existing dataset or snapshot (multiple values, separated by commas)
policy	QOS policy

qos.policy.assign_pool ADMIN

CHANGE

Description Assign pool to policy.

Syntax

```
qos.policy.assign_pool policy=QOSPOLICY pool=POOL[,POOL,...]
```

Arguments

pool	Name of an existing pool (multiple values, separated by commas)
policy	QOS policy

qos.policy.create ADMIN

CHANGE

Description Create a new policy.

Syntax

```
qos.policy.create name=NAME dataset_type=Dataset type [max_ops=FLOAT] [max_mbps=MAXBPS]
[burst=YESNO] [burst_ratio=INT] [duration_at_full_burst=INT]
```

Arguments

name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@()[]$=!-#{}%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
max_mbps	A positive number (up to 1 digit beyond the decimal point)
max_ops	A number
dataset_type	Volumes or Filesystems
burst	Either yes, or no
burst_ratio	An integer number
duration_at_full_burst	An integer number

qos.policy.delete ADMIN

CHANGE

Description Delete a policy.

Syntax

```
qos.policy.delete policy=QOSPOLICY
```

Arguments

policy	QOS policy

qos.policy.entity_query ALL ROLES

CHANGE

Description List assosiated entities.

Syntax

```
qos.policy.entity_query [policy=QOSPOLICY] [dataset_type=Dataset type]
```

Arguments

dataset_type	Volumes or Filesystems
policy	QOS policy

qos.policy.modify ADMIN

CHANGE

Description Modify a policy.

Syntax

```
qos.policy.modify policy=QOSPOLICY [new_name=NAME] [max_ops=FLOAT] [max_mbps=MAXBPS] [burst=YESNO] [burst_ratio=INT] [duration_at_full_burst=INT]
```

Arguments

policy	QOS policy
new_name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
max_ops	A number
max_mbps	A positive number (up to 1 digit beyond the decimal point)
duration_at_full_burst	An integer number
burst_ratio	An integer number
burst	Either yes, or no

qos.policy.query ALL ROLES

CHANGE

Description List existing policies.

Syntax

```
qos.policy.query [policy=QOSPOLICY] [dataset_type=Dataset type]
```

Arguments

dataset_type	Volumes or Filesystems
policy	QOS policy

qos.policy.remove_dataset ADMIN

CHANGE

Description Remove dataset from policy.

Syntax

```
qos.policy.remove_dataset dataset={DATASET|SNAP} policy=QOSPOLICY
```

Arguments

dataset	Name of an existing dataset or snapshot
policy	QOS policy

qos.policy.remove_pool ADMIN

CHANGE

Description Remove pool from policy.

Syntax

```
qos.policy.remove_pool pool=POOL policy=QOSPOLICY
```

Arguments

`pool`	Name of an existing pool
`policy`	QOS policy

qos_policy.assign ADMIN

CHANGE

Description **Assign entity to policy.** Each policy is effective to only one type of entity:

- Volume
- Filesystem
- Pool-volume - the policy impacts the volumes that belong to this pool
- Pool-filesystem - the policy impacts the filesystems that belong to this pool

Syntax

```
qos_policy.assign policy=QOSPOLICY [vol={VOL|SNAP}[,{VOL|SNAP},...]] [fs={FS|SNAP}[,{FS|SNAP},...]]
[pool=POOL[,POOL,...]]
```

Arguments

`pool`	Name of an existing pool (multiple values, separated by commas)
`fs`	Name of an existing filesystem or snapshot (multiple values, separated by commas)
`vol`	Name of an existing volume or snapshot (multiple values, separated by commas)
`policy`	QoS policy

Assigning a volume to a policy

Example

```
qos_policy.assign policy=q1 vol=v1
```

Output

```
VOLUME 'v1' was assigned to policy 'q1'
```

Assigning a filesystem to a policy

Example

```
qos_policy.assign policy=q2 fs=fs1
```

Output

```
FILESYSTEM 'fs1' was assigned to policy 'q2'
```

Assigning all volumes within a pool to a policy

Example

```
qos_policy.assign policy=q4 pool=p1
```

Output

```
POOL 'p1' was assigned to policy 'q4'
```

Assigning all filesystems within a pool to a policy

Example

```
qos_policy.assign policy=q3 pool=p1
```

Output

```
POOL 'p1' was assigned to policy 'q3'
```

qos_policy.assignment_query ALL ROLES

CHANGE

Description List entities that are associated with a policy.

Syntax

```
qos_policy.assignment_query [policy=QOSPOLICY] [type=QoS policy type]
```

Arguments

type	VOLUME, FILESYSTEM, POOL_FILESYSTEM or POOL_VOLUME
policy	QoS policy

Example

```
qos_policy.assignment_query
```

Output

NAME	TYPE	POLICY	ENTITIES
v1	VOLUME	q1	-
fs1	FILESYSTEM	q2	-
p1	POOL_VOLUME	q4	1
p1	POOL_FILESYSTEM	q3	1

qos_policy.create ADMIN

CHANGE

Description Create a new QoS policy.

Syntax

```
qos_policy.create name=NAME type=QoS policy type [max_ops=QOSOPS] [max_throughput=MAXMIBPS]
[burst=BURST] [burst_factor=BURSTFACTOR]
```

Arguments

name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
burst_factor	
burst	Enable or disable Burst for QoS policy
max_throughput	QoS policy max throughput in MiB/s
max_ops	
type	VOLUME, FILESYSTEM, POOL_FILESYSTEM or POOL_VOLUME

Creating a policy with max IOPS and no burst

Example

```
qos_policy.create name=q5 type=volume max_ops=2000
```

Output

```
QoS policy "q5" created
```

Creating a policy with max IOPS, and a burst

Example

```
qos_policy.create name=q6 type=volume max_ops=2000 burst_factor=4 burst=yes
```

Output

```
QoS policy "q6" created
```

qos_policy.delete ADMIN

CHANGE

Description Delete a policy.

- There is no need to unassigned entities from a policy prior to the deletion
- Entities that were assigned to this policy are no longer assigned to a policy

Syntax

```
qos_policy.delete policy=QOSPOLICY
```

Arguments

policy	QoS policy

Example

```
qos_policy.delete policy=q1
```

Output

```
QoS policy "q1" deleted
```

qos_policy.modify ADMIN

CHANGE

Description Modify a policy.

- Rename the policy
- Change the max operations or max throughput settings
- Add a burst
- Activate/deactivate the burst

Syntax

```
qos_policy.modify policy=QOSPOLICY [new_name=NAME] [max_ops=QOSOPSORNONE]
[max_throughput=MAXMIBPSORNONE] [burst=BURST] [burst_factor=BURSTFACTOR]
```

Arguments

policy	QoS policy
new_name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
max_ops	
max_throughput	NONE, or qoS policy max throughput in MiB/s
burst_factor	
burst	Enable or disable Burst for QoS policy

Renaming a policy

Example

```
qos_policy.modify policy=q1 new_name=q2
```

Output

```
Policy "q1" renamed to "q2"
```

Changing the max operations and max throughput parameters

Example

```
qos_policy.modify policy=q1 max_ops=3000
```

Output

```
QoS policy "q1" updated
```

Changing the burst factor

Example

```
qos_policy.modify policy=policy-1 burst_factor=4 burst=yes
```

Output

```
QoS policy "policy-1" updated
```

Deactivating the burst

Example

```
qos_policy.modify policy=policy-1 burst=no
```

Output

```
QoS policy "policy-1" updated
```

qos_policy.query ALL ROLES

CHANGE

Description List existing policies.

Syntax

```
qos_policy.query [policy=QOSPOLICY] [type=QoS policy type]
```

Arguments

type	VOLUME, FILESYSTEM, POOL_FILESYSTEM or POOL_VOLUME
policy	QoS policy

Example

```
qos_policy.query
```

Output

NAME	TYPE	MAX OPS	MAX THROUGHPUT	BURST FACTOR
q1	VOLUME	3000	UNLIMITED	4
q2	FILESYSTEM	3000	UNLIMITED	-
q3	POOL_VOLUME	10000	UNLIMITED	-
q4	POOL_FILESYSTEM	10000	UNLIMITED	-

qos_policy.unassign ADMIN

CHANGE

Description **Remove entity from policy.** Unassigning an entity from a policy is done in either of the following ways:

- Specifying the entity
- Specifying the policy and then the entity

It is impossible to unassign all of the entities of a policy.

Syntax

```
qos_policy.unassign [vol={VOL|SNAP}[,{VOL|SNAP},...]] [fs={FS|SNAP}[,{FS|SNAP},...]]
[pool=POOL[,POOL,...]] [policy=QOSPOLICY]
```

Arguments

vol	Name of an existing volume or snapshot (multiple values, separated by commas)
policy	QoS policy
pool	Name of an existing pool (multiple values, separated by commas)
fs	Name of an existing filesystem or snapshot (multiple values, separated by commas)

Specifying the entity.

Example

```
qos_policy.unassign vol=v1
```

Output

```
Entity was unassigned from QoS policy
```

Specifying the policy and the entity.

Example

```
qos_policy.unassign policy=q1 vol=v1
```

Output

```
Entity was unassigned from QoS policy
```

replica.change_type ADMIN POOL ADMIN CHANGE

Description **Change replica type**

THis command changes the type of the replica from synchronous to asynchronous and from asynchronous to synchronous.

Changing the type of the replica from asynchronous to synchronous

- The source system will initiate a sync job. Until the sync job is finished, the sync state will be SYNC_IN_PROGRESS

Changing the type of the replica from synchronous to asynchronous:

- The RPO and Interval parameters are mandatory

Syntax

```
replica.change_type local_dataset=REPLICATEDENTITY [dataset_type=CGORDATASET] [rpo=TIMEDELTA] [interval=TIMEDELTA]
```

Arguments

`interval`	A time delta, e.g: '02:02', '05:11:06'
`rpo`	A time delta, e.g: '02:02', '05:11:06'
`dataset_type`	CG, VOLUME or FILESYSTEM
`local_dataset`	Name of an existing dataset or consistency group with replication

replica.member_query ALL ROLES CHANGE

Description **List members of a replicated consistency group.**

The members are displayed along with their initialization status and restore point, so the replica consistency state can be viewed in a glance.

Syntax

```
replica.member_query local_cg=OLDREPLICATEDCG
```

Arguments

local_cg	Name of an existing consistency group with replication

In the following example, querying for the members of cg-1 results in a list of local datasets and their remote replicants.

Example

```
replica.member_query local_cg=cg-1
```

Output

LOCAL DATASET	REMOTE DATASET	INITIALIZING	RESTORE POINT
vol-1	vol-1-target	no	2016-05-31 10:00:00
vol-2	vol-2-target	no	2016-05-31 10:00:00

system.cod_query ADMIN

CHANGE

Description Query the system for Capacity On Demand (COD) consumption.

Syntax

```
system.cod_query
```

Run the command with no arguments to display the detailed capacity consumption per pool.

Example

```
system.cod_query
```

Output

POOL	CONSUMED COD
pool1	100.00 TB
pool2	200.00 TB
---	---
TOTAL	300.00 TB

Run the command with the --detailed argument to get the system total capacity consumption.

Example

```
system.cod_query
```

Output

```
300.00 TB
```

user.map_role

CHANGE

Description Map LDAP or Active Directory group to user role.

The mapped user roles are displayed with the pools they have access to.

Syntax

```
user.map_role
```

Example

```
user.map_role ldap=infinidat.com group=CN=Administrators,CN=Builtin,DC=infinidat,DC=com role=Admin
```

Output

```
LDAP server AD group Administrators mapped to role ReadOnly
```

user.reset_password

CHANGE

Description Send password reset email to local user.

Follow the instructions on the email.

Syntax

```
user.reset_password
```

Example

```
user.reset_password name=user1
```

Output

```
Email sent to user1
```

user.unmap_role

CHANGE

Description Unmap LDAP or Active Directory group from user role.

Syntax

```
user.unmap_role
```

Example

```
user.unmap_role ldap=infinidat.com group=CN=Administrators,CN=Builtin,DC=infinidat,DC=com
```

Output

vol.assign_to_qos_policy ADMIN

CHANGE

Description Assign volume to policy.

Syntax

```
vol.assign_to_qos_policy vol=VOL policy=QOSPOLICY
```

Arguments

policy	QOS policy
vol	Name of an existing volume

vol.clone.cache ADMIN POOL ADMIN

CHANGE

Description Enable or disable SSD cache.

Set the optional parameter Recursive to Yes in order to apply the SSD cache settings to the entire clone tree. The clone tree means the descendants of the clone only. The clone ancestors are not included.

Syntax

```
vol.clone.cache clone={VOL|SNAP} ssd_cache=YESNO [recursive=YESNO]
```

Arguments

recursive	Either yes, or no
ssd_cache	Either yes, or no
clone	Name of an existing volume or snapshot

Example

```
vol.clone.cache name=clone1 ssd=yes
```

Output

```
SSD cache enabled
```

vol.clone.create ADMIN POOL ADMIN

CHANGE

Description Create a new volume clone from volume snapshot.

Syntax

```
vol.clone.create snap={VOL|SNAP} name=NAME [ssd_cache=YESNO]
```

Arguments

name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@(){}[]$=!-#%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
snap	Name of an existing volume or snapshot
ssd_cache	Either yes, or no

Example

```
vol.clone.create snap=snap1 name=clone1
```

Output

```
Volume clone clone1 created
```

vol.clone.delete ADMIN POOL ADMIN

CHANGE

Description Delete volume clone.

Syntax

```
vol.clone.delete clone={VOL|SNAP}[,{VOL|SNAP},...]
```

Arguments

clone	Name of an existing volume or snapshot (multiple values, separated by commas)

Example

```
vol.clone.delete name=clone1
```

Output

```
Volume clone clone1 deleted
```

vol.clone.map ADMIN POOL ADMIN

CHANGE

Description Map volume clone to host or cluster.

Either the host, cluster or LUN must be stated.

Syntax

```
vol.clone.map clone={VOL|SNAP}[,{VOL|SNAP},...] [host=HOST] [cluster=CLUSTER] [lun=LUN]
```

Arguments

host	Name of an existing host
clone	Name of an existing volume or snapshot (multiple values, separated by commas)
cluster	Name of an existing cluster
lun	SCSI logical unit number (LUN)

Example

```
vol.clone.map name=clone1 cluster=cluster1
```

Output

```
Volume clone clone1 mapped to LUN LUN1 in cluster cluster1
```

vol.clone.map_query ALL ROLES

CHANGE

Description List host and cluster mappings by volume.

Syntax

```
vol.clone.map_query [clone={VOL|SNAP}[,{VOL|SNAP},...]] [host=HOST] [cluster=CLUSTER]
```

Arguments

cluster	Name of an existing cluster
host	Name of an existing host
clone	Name of an existing volume or snapshot (multiple values, separated by commas)

Example

```
vol.clone.map_query
```

Output

NAME	TARGET TYPE	TARGET NAME	LUNS
clone1	CLUSTER	cluster1	11

vol.clone.query ALL ROLES

CHANGE

Description List existing clones.

The output of this command displays all of the clones on InfiniBox. Use the unit argument to specify the units of the capacity fields of the output.

Syntax

```
vol.clone.query [clone={VOL|SNAP}[,{VOL|SNAP},...]] [snap={VOL|SNAP}] [cg=CG] [pool=POOL]
[unit=CAPACITYUNIT] [mapped=YESNO]
```

Arguments

snap	Name of an existing volume or snapshot
clone	Name of an existing volume or snapshot (multiple values, separated by commas)
mapped	Either yes, or no
unit	B, G (or GB), GiB, T (or TB), TiB or block
pool	Name of an existing pool
cg	Name of an existing consistency group

Example

```
vol.clone.query
```

Output

NAME	SIZE	THIN	POOL	WP	MAPPED	SSD CACHE	ALLOCATED	REFERENCED	TREE ALLOCATED	CREATED AT
clone1	1TB	yes	pool1	yes	yes	yes	1TB	0TB	1TB	2014-05-11 10:00:00

vol.clone.rename ADMIN POOL ADMIN

CHANGE

Description Rename volume clone.

Syntax

```
vol.clone.rename clone={VOL|SNAP} new_name=NAME
```

Arguments

new_name	A maximum of 65 Latin characters, numbers, spaces, and the following symbols: "^&'@()[]$=!-#{}%.+~_" (excluding quotation marks). Leading and trailing whitespace characters are stripped.
clone	Name of an existing volume or snapshot

Example

```
vol.clone.rename name=clone1 new_name=clone2
```

Output

```
Volume clone clone1 renamed to clone2
```

vol.clone.resize ADMIN POOL ADMIN

CHANGE

Description Resize volume clone.

Syntax

```
vol.clone.resize clone={VOL|SNAP} size=SIZEDELTA
```

Arguments

size	A number (up to 2 digits beyond the decimal point), optionally followed by a unit; e.g. 1TB, 1000000000000. When preceded by a plus or minus sign, represents a delta; e.g. 3000000m (absolute size), +2000GB (positive delta), -1.5T (negative delta)
clone	Name of an existing volume or snapshot

Example

```
vol.clone.resize name=clone1 size=10g
```

Output

```
Volume clone clone1 updated
```

vol.clone.restore ADMIN POOL ADMIN

CHANGE

Description Restore data state of a volume clone from one of its snapshots.

This command restores a clone. To restore a volume, use the vol.restore command. The clone can be restored from any of its snapshots or clones.

Syntax

```
vol.clone.restore clone={VOL|SNAP} source=SNAP
```

Arguments

source	Name of an existing volume snapshot
clone	Name of an existing volume or snapshot

Example

```
vol.clone.restore name=clone1 snap=snap1
```

Output

```
Volume clone clone1 restored from snapshot snap1
```

vol.clone.tree ALL ROLES

CHANGE

Description Display the hierarchy of clones and their descendants.

Syntax

```
vol.clone.tree clone={VOL|SNAP} [unit=CAPACITYUNIT]
```

Arguments

clone	Name of an existing volume or snapshot
unit	B, G (or GB), GiB, T (or TB), TiB or block

Note that in the following example, the snapshot is a descendant of the clone.

Example

```
vol.clone.tree
```

Output

NAME	TYPE	SIZE
clone1	CLONE	1TB
|----snap1	SNAP	1TB

vol.clone.unmap ADMIN POOL ADMIN

CHANGE

Description Unmap volume clone from host or cluster.

Syntax

```
vol.clone.unmap clone={VOL|SNAP}[,{VOL|SNAP},...] [host=HOST] [cluster=CLUSTER]
```

Arguments

cluster	Name of an existing cluster
host	Name of an existing host
clone	Name of an existing volume or snapshot (multiple values, separated by commas)

Example

```
vol.clone.unmap name=clone1 cluster=cluster1
```

Output

```
Volume clone clone1 unmapped from LUN LUN1 in cluster cluster1
```

vol.clone.write_enable ADMIN POOL ADMIN

CHANGE

Description Enable write access on volume clone.

Set the optional parameter Recursive to Yes in order to enable writing to the entire clone tree.

Syntax

```
vol.clone.write_enable name={VOL|SNAP} [recursive=YESNO]
```

Arguments

recursive	Either yes, or no
name	Name of an existing volume or snapshot

Example

```
vol.clone.write_enable name=clone1
```

Output

```
Volume clone clone1 write access enabled
```

vol.clone.write_protect ADMIN POOL ADMIN

CHANGE

Description Disable write access on volume clone.

Set the optional parameter Recursive to Yes in order to disable writing to the entire clone tree.

Syntax

```
vol.clone.write_protect clone={VOL|SNAP} [recursive=YESNO]
```

Arguments

clone	Name of an existing volume or snapshot
recursive	Either yes, or no

Example

```
vol.clone.write_protect name=clone1
```

Output

```
Volume clone clone1 write access disabled
```

vol.remove_from_qos_policy ADMIN

CHANGE

Description Remove volume from policy.

Syntax

```
vol.remove_from_qos_policy vol=VOL policy=QOSPOLICY
```

Arguments

policy	QOS policy
vol	Name of an existing volume

vol.snap.assign_to_qos_policy ADMIN

CHANGE

Description Assign volume to policy.

Syntax

```
vol.snap.assign_to_qos_policy snap=SNAP policy=QOSPOLICY
```

Arguments

snap	Name of an existing volume snapshot
policy	QOS policy

vol.snap.group_create

CHANGE

Description Create a consistent set of snapshots of multiple volumes or clones.

Select multiple sources - volumes and clones - from the same pool. Run vol.tree or vol.query to view the results. You cannot select snapshots as an input for this command.

Syntax

```
vol.snap.group_create
```

Example

```
vol.snap.group_create source=vol1,vol2,vol3 snap_prefix=sgroup1-
```

Output

```
Consistent snapshots created
```

vol.snap.remove_from_qos_policy ADMIN

CHANGE

Description Remove volume from policy.

Syntax

```
vol.snap.remove_from_qos_policy snap=SNAP policy=QOSPOLICY
```

Arguments

`snap`	Name of an existing volume snapshot
`policy`	QOS policy

Chapter 45. Filesystem Clone