Vormetric Security Server CLI

Version 6.0

Release Date 12/28/2016

VMSSC Reference

Thales/Vormetric

Table of Contents

Contents

Overview

VMSSC (Vormetric Security Server CLI) is a command line interface to the Data Security Manager (DSM), which is also referred to as the security server .VMSSC can do things like automate deployments, script routine and repetitive tasks, and perform unattended batch processing.

General Command-Line Structure

The command structure of VMSSC is modeled after the way Concurrent Versions System (CVS) command operates. The following is an example of the CVS command structure:

```
Usage: cvs [cvs-options] command [command-options-and-arguments]
```

Global options precede the command. Command-specific options follow the command.The following is an example of the general format of all VMSSC commands:

```
vmssc [auth options] command_group command <command-options-and-arguments>
```

- The **[auth options]** are the same for every command_group and command. They exist so that the authentication information, like username, password, and domain, can be entered on a per-command basis.
- The **command_group** specifies the top-level category to operate upon. Currently the types are "host", "group", "server", "policy", "key", "domain", and "user".
- The **command** represents some set of commands that apply to the command_group. Typical commands are "add", "delete", "modify", and "show".
- **command-options-and-arguments** are specified for each command.

We describe the command_group followed by the command, for example "./vmssc host show". The reverse order is also acceptable. For example, "./vmssc show host".

The VMSSC command groups are as follows:

Name	Description
Server Commands	Shows basic information about the server.
Host Commands	Actions involving the creation/modification/destruction life cycle of hosts. Also adding and removing GuardPoints.
Group Commands	Similar to hosts, this applies to host groups. Hosts can be entered into and removed from groups.
Policy Commands	Actions involving the creation, modification, and removal of policies.
Key Commands	Actions involving the creation, modification, and removal of keys. This is used to operate on both "agent keys" (those used by our products) and "vault keys" (those that we store only).
Admin Commands	Actions involving the creation and maintenance of users (also known as administrators), adding and removing them from domains, etc.
Domain Commands	Actions involving the creation, modification, and removal of domains.
Certificate Commands	Actions involving the storage, searching, retrieval, and removal of certificates.
Signature Commands	Actions involving signature sets and creating signatures.
Report Commands	Actions involving reports.
Rekey Commands	Actions involving rekeying operations and control of live data transform.
Syslog Commands	Actions involving syslog servers.
Backup Commands	Actions involving backups, viz. initializing, deleting, visualizing backups.
Sharedsecret Commands	Actions involving shared secret and default license type, viz. adding, deleting, visualizing registration shared secret/default license type.

Getting Help

You can get help by typing `vmssc` with no arguments.

```
Usage: vmssc [auth_options] <object> <command> [cmd_options] [target]
  where auth_options are -c, -d, etc.
    specify --help-auth-options for a list of options.
  where object is host, group, etc.
    specify --help-objects for a list of objects.
  where command is add, del, etc.
    specify command 'help' for object-specific help.
  Specify --help to see this message.
```

Use the following examples to get help on all parts of the command line syntax:

- `vmssc --help-auth-options` shows the authentication options to the Security Server
- `vmssc --help-objects` shows the list of command groups available
- `vmssc help <command group>` shows help for that particular command group. For example, `vmssc help host` describes the command line syntax for host operations.

Return Codes

Exit Value	Meaning
0	Success
1	Problem parsing the command line, vmssc.conf file, etc
2	Problem communicating with the security server
3	Bad username/password, or user does not have permission on security server
4	Other error from security server

When a failure occurs, an informative reason is always printed to `stderr`.

Authentication Options

The general format of all VMSSC commands is

```
vmssc [auth options] command_group command <command-options-and-arguments>
```

The **[auth options]** are the same for every command_group and command. They exist so that the authentication information, like username, password, local domain name or domain name, can be entered on a per-command basis. This information can also be cached in a file; see below.

There are many ways to provide authentication options, including through a config file, from the environment, and on the command line. This table describes how each authentication component is supplied.

Option in Config File	Option in Environment	Option on the Command Line	Description
server	VMSSC_SERVER	-s or --server	The target server (a.k.a. DSM). E.g. A valid ipv6 address can be specified as [fd01::3:15:130]. It is mandatory to write an ipv6 address in between square brackets otherwise it would be treated as "ipv4 address : Port number".
username	VMSSC_USERNAME	-u or --username	The administrator on that server
password	VMSSC_PASSWORD	-p or --password	The password for the server or administrator. This can also be entered using an echoless prompt.
domain	VMSSC_DOMAIN	-d or --domain	The domain in which the command should be run (note: case sensitive!). Use of contradicting local domain name/global domain name is not allowed. E.g. Use of different global domain name and local domain name at same time is not allowed.
localdomain	VMSSC_LOCALDOMAIN	-l or –locdom	The local domain name in which a local domain/security administrator is logging in.

defaulthostpassword	VMSSC_DEFAULTHOSTPASSWORD	(Part of host commands)	The default value of the host password (the host add command only)
cacert	VMSSC_CACERT	-c or --ca-file	Path to a server CA certificate. This option is added to verify the authenticity of the server. It should point to a file in .pem format that contains the CA (the root of trust) certificate of the server. If it is not provided, the server's identity will not be verified and we may be subject to man-in-the-middle attacks (This is just like ignoring the warnings in a web browser about a questionable certificate). The most direct and practical way to acquire the certificate of the CA is to register a host and then go to /opt/vormetric/DataSecurityExpert/agent/vmd/pem and copy the file “agent_signer-cert.pem
timeout	VMSSC_TIMEOUT	-t or --timeout	How long to wait for a response (seconds). Default is 10 minutes. A value of zero means infinite or no timeout.
port	VMSSC_PORT	-r or --port	Specify port on which VMSSC wants to communicate with DSM. Valid values are 8445, 8448(for Suite B) and 443. This is not a required parameter. If not specified, then port number 443 will be used as default port.

The following list represents the order in which information is processed; later information supersedes prior information. Therefore, command line information supersedes everything else.

1. From configuration files:
    1. `~/.vmsscpass` That's the file `.vmsscpass` in the user's home directory (%APPDATA% in windows). This file can only be created by "`vmssc server login`", as described below.
    2. `~/vmssc.conf` Again, this in the user's home directory.
    3. `vmssc.conf` The same file name, but this time in the current working directory
    4. `$VMSSC_CONF` If the `VMSSC_CONF` environment variable is set, it's value is assumed to be a file name specifying a config file.
2. From the environment
3. On the command line

Configuration files all share a common format. They are made up of "general attribute=value" pairs. For example:

```
username = <username>
```

```
password = <password>
domain = <domain>
server = <server_name>
defaulthostpassword = <hostpass>
cacert = </path/to/ca_cert.pem>
```

Similarly, a local domain user can have the following fields for logging into his domain:

```
username = <username>
password = <password>
localdomain = <local_domain_name>
server = <server_name>
defaulthostpassword = <hostpass>
cacert = </path/to/ca_cert.pem>
```

It is to be noted that a local user willing to login into his local domain need to include his domain name as localdomain=<domain_name> instead of domain name in the ~/vmssc.conf file or into environment variable as specified in the table above. This option is given for providing multi-tenancy support to the VMSSC command group.

A file like this can be placed in `~/vmssc.conf`, `vmssc.conf`, or the value of `$VMSSC_CONF`.
The `~/.vmsscpass` file follows similar rules, but can only be created by "`vmssc server login`", again described below. Next processed, are environment variables. The names of the environment variables are the same as those for the configuration files, but prepended with VMSSC_ and in all caps. Example:

```
export VMSSC_PASSWORD = <clever-password>
```

Putting all this together, it's possible to use a mixture of files, environment, and command line options for the [authentication options]. For example, it's possible to have the following contents of `~/vmssc.conf`:

```
username = barney
domain = domain1
server = testdsm3.i.vormetric.com
defaulthostpassword = xxxyyy123
cacert = /path/to/ca_cert.pem
```

And the following in `vmssc.conf`:

```
server = good2.go.com
```

Then in the environment

```
VMSSC_PASSWORD=Abc1234#
```

Finally on the command line

```
./vmssc -u voradmin server show
```

The algorithm for looking at these options is as follows:

1. Try to open `~/.vmsscpass` : it doesn't exist, so move on.
2. `Try to open ~/vmssc.conf: It exists, and its contents are read. The attributes are now`

```
username = barney
domain = domain1
server = testdsm3.i.vormetric.com
defaulthostpassword = xxxyyy123
cacert = /path/to/ca_cert.pem
```

3. Try to open `vmssc.conf`: It exists, and its contents are read. The attributes are now

```
username = barney
domain = domain1
server = good2.go.com
defaulthostpassword = xxxyyy123
cacert = /path/to/ca_cert.pem
```

4. Get the VMSSC_CONF environment variable: The .conf file does not exist. The next highest authentication criteria is determined by environmental variables, if any.
5. Read environment variables. VMSSC_PASSWORD exists, so use it. The attributes are now

```
username = barney
domain = domain1
server = good2.go.com
defaulthostpassword = xxxyyy123
cacert = /path/to/ca_cert.pem
password = Abc1234#
```

6. Parse command line options. The -u option is found and applied. The attributes are now

```
username = voradmin
domain = domain1
server = good2.go.com
defaulthostpassword = xxxyyy123
cacert = /path/to/ca_cert.pem
password = Abc1234#
```

With so many input mechanisms, the source of these values may cause some confusion. For troubleshooting purposes, the "-W" (or "-which") flag has been provided. When invoked, this flag shows the interesting attributes, their values, and where they come from. For the previous example, the output is:

```
$ ./vmssc -u voradmin -W
Required parameter SERVER (good2.go.com) obtained from the file vmssc.conf
Required parameter USERNAME (voradmin) obtained from the command line
Required parameter PASSWORD (Abc1234#) obtained from the environment
Required parameter DOMAIN (domain1) obtained from the file
/home/myspace/vmssc.conf
Required parameter LOCALDOMAIN (domain2) obtained from the environment
File /home/myspace/.vmsscpass does not exist
VMSSC_CONF is not set in the environment
```

Using the "`vmssc server login`" command is an alternate way of providing this sort information by caching credentials on the local machine. It is described in the Server Commands section. If all of the required parameters are available except password, then the password will be prompted for as follows:

```
$ ./vmssc -s toaster -u User1 server show
```

```
Password:
```

Server Commands

The server commands include the following:

Name	Description
login	Caches credentials for authentication to the DSM
logout	Removes cached credentials
show	Shows basic information about the server
license	Upload a license file to the DSM

Detailed explanations and examples of the Server Commands are given in the next sections.

Server Login

The"login" command stores a set of credentials for repeated use. The "logout" command removes the stored set of credentiales. In the following example, the options are set using command line options, there are no configuration files present:

```
$ ./vmssc -s test-dsm.vormetric.com -u User1 -p User1234! -d Space server
login
$ ./vmssc server show
Server Name    : test-dsm.vormetric.com
Version        : 5.0.3.754
$ ./vmssc server logout
```

The previous example illustrates:

1. The "server login" command verifies that the servername/username/password permits access into the server.
2. The command creates the `~/.vmsscpass` file and puts this information into it in an encrypted or obfuscated format. Future calls make use of the ~/.`vmsscpass` file.
3. The "server logout" command removes the `~/.vmsscpass` file.
4. By default, .vmsscpass will be stored in the location according to HOME or APPDATA (for windows) environment variable. If you want to change the default location, set VMSSC_HOME to other directory.

The `~/.vmsscpass` file is shown below:

```
$ cat ~/.vmsscpass
SERVER_OBFUSCATED = 5F538EABF11B11B8A1D98A7FECE02A78
USERNAME_OBFUSCATED = 155AA921F2F9CE823870D99979332A8A
DOMAIN_OBFUSCATED = CB69973EBB86D34FCE272442CF988721
PASSWORD_OBFUSCATED = B9BA02CBCB10079E3C2463AAD0BBB08A
PORTIBILITY_OBFUSCATED =
471C98D9E8490AF156B0649932217C92A016222317328BF46EDD95EDDA218924
EXPIRY_OBFUSCATED = 6DD48275F9C8039FB3317C09D643CD2D
```

```
DIGEST =
b1c0ffe09383b01cf60fa0e67dfdc2cd151673e89d2ff7d84ee2cefeafd565b3bb1078b080c8c
d154be041102594395d731bf700acb77e5dec385b1af9514e10
```

Two arguments can be used with the "server login" command.

Short Form	Long Form	Description
-r	--redistributable	Normally, the `~/.vmsscpass` file will only work on the machine that it was created on. That is, if the file is transfered to another machine it won't work. However, if this flag is specified, the `~/.vmsscpass` file can work on any machine.
-x <expiry>	--expiry <expiry>	Normally, the `~/.vmsscpass` file is valid for one day. Use the `-x` argument to change that. The <expiry> option must be of the form nnnX, where the n represents a numeric digit and X represents the units in m (minutes), h (hours), or d (days). If X is omitted the units are assumed to be hours. Examples: 12m (12 minutes), 5h (5 hours), 17d (17 days), 6 (6 hours). A value of 0 (zero) means that there is no expiration, and the `~/.vmsscpass` file can be used indefinitely.

Server Logout

This command removes the ~/.vmsscpass file that was created by the server login command.

Server Show

```
vmssc server show [options]
```

The following options are supported:

Short Option	Long Option	Description
-d	--detailed	Display detailed server information
-f	--failover	Display failover server status
-h <server>	--hosts <server>	Display hosts assigned to server
-u	--usage	Display appliance usage statistics
-a	--all	Equivalent to -d -f -u

With no options specified, this command option displays basic server information:

```
$ ./vmssc server show
$ ./vmssc server show
Server Name : agenttest100.i.vormetric.com
Version     : 5.1.0.808
```

The **-d** option shows detailed server information:

```
$ ./vmssc server show -d
Server Name    : abctest100.i.vormetric.com
Version        : 5.1.0.808
```

```
Time           : 2012-09-19 16:50:07.293 PDT
Fingerprint    : 5E:50:DA:9A:BF:0D:3E:9A:3C:A4:B3:3F:45:BF:E4:C3:08:07:4D:A4
EC Fingerprint : 67:E0:4E:77:F0:6C:12:23:29:41:5F:0D:03:1C:8B:FC:7C:D5:A8:07
HA Serv[0]     : abctest100.i.vormetric.com Primary CONFIGURED 0
License Header : Agt  Type    Quota  Usage  Expiry
License[0]     : DB2  TERM    100    0      Sep-30-2012
License[1]     : IDS  TERM    100    0      Sep-30-2012
License[2]     : FS   TERM    100    0      Sep-30-2012
License[3]     : KEY  TERM    1000   2      Sep-30-2012
License[4]     : DB2  HOURLY  100    0      -
License[5]     : IDS  HOURLY  0      0      -
License[6]     : FS   HOURLY  0      0      -
License[7]     : KEY  HOURLY  0      0      -
```

The **-f** option shows failover server information, in particular the status of the replication between the failover and primary nodes:

```
$ ./vmssc server show -f
test-dsm.vormetric.com : HA Role              : Primary
test-fov.vormetric.com : HA Role              : Failover
test-fov.vormetric.com : HA Configured        : Yes
test-fov.vormetric.com : HA Registered        : Yes
test-fov.vormetric.com : HA Last run time     : 2012-09-19 16:51:21.111
test-fov.vormetric.com : HA Last sync time    : 2012-09-19 16:51:21.111
test-fov.vormetric.com : HA Last sync status  : Success
test-fov.vormetric.com : HA Last sync code    : 0
test-fov.vormetric.com : HA Last sync message :
```

The **-u** option shows statistics regarding the resource consumption of the physical appliance box. Information on the load average, memory, and disk consumption are reported.

```
----- Uptime Information -----
 16:10:14 up  2:12,  1 user,  load average: 0.53, 0.51, 0.54

----- Filesystem Information -----
Filesystem           1M-blocks      Used Available Use% Mounted on
/dev/sda6                 9389      2987      5925  34% /
/dev/sda9               250798      4755    233304   2% /partitions/large
/dev/sda1                  891        22       824   3% /grub
tmpfs                     1963         0      1963   0% /dev/shm
/dev/sda2                 9387      2996      5915  34% /partitions/std/2
/dev/sda8                 7505       155      6970   3% /tmp

----- Virtual Memory Information -----
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu--
----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa
st
 1  0     76 159600  68896 3156572    0    0    39   209  258  524  3  0 95
1  0
```

The "Virtual Memory Information" section is the output of the unix "vmstat" call. The columns are as follows:

- Procs

    - r: The number of processes waiting for run time.
    - b: The number of processes in uninterruptable sleep.
- Memory
    - swpd: the amount of virtual memory used (kB).
    - free: the amount of idle memory (kB).
    - buff: the amount of memory used as buffers (kB).
    - cache: the amount of memory used as cache (kB)
- swap
    - si: Amount of memory swapped in from disk (kB/s).
    - so: Amount of memory swapped to disk (kB/s).
- IO
    - bi: Blocks sent to a block device (blocks/s).
    - bo: Blocks received from a block device (blocks/s).
- System
    - in: The number of interrupts per second, including the clock.
    - cs: The number of context switches per second.
- CPU
    - us: user time
    - sy: system time
    - id: idle time
    - wa: Time spent waiting for IO
    - st: Time stolen from a virtual machine

The "**-h <server>**" option shows the hosts assigned to the given primary or failover server.

```
$ ./vmssc server show -h testdsm3.i.vormetric.com
testhosts.vormetric.com
anyuser1-xp.vormetric.com
```

Finally, the **-a** option shows the **-d, -f,** and **-u** output, in that order.

Server License

This command uploads a license file to the DSM. It takes one mandatory argument: **-f**

Short Option	Long Option	Description
-f <filename>	--file <filename>	Mandatory argument. The file specified must be a valid license file. The DSM begins using the new license file and the features it specifies immediately.

Example:

```
 $ ./vmssc server license -f /path/to/license/file
```

Host Commands

Valid host commands are:

Host Command	Description
add	adds a host to the server
modify	modifies attributes of an existing host
delete	deletes a host from the server
addgp	adds a GuardPoint to an existing host
modgp	modifies a GuardPoint on an existing host
delgp	removes a GuardPoint from an existing host
show	displays host information
docker	displays host's docker information.
showgp	Shows Guard point information.

Host Add

```
vmssc host add [opts] hostname
```

This command adds a host to the server. The permitted options are:

Argument	Long Form	Short Description	Notes
-a	--autoassign	A host will be auto-assigned to an HA server	Default behavior (flag not specified) will leave the host assigned to the primary. This is incompatible with -s, which will make the assignment manually.
-d "Description"	--description	A description of this host.	Don't forget to quote multi-word strings for the shell.
-p port	--port	The port the host uses.	Defaults to 7024.
-g hostgroup	--group	Add the host to this hostgroup.	Synonym: --hostgroup
-h hostpass	--hostpass	Set the host password.	Mutually exclusive with -G. Synonym: --password
-G	--generate	Generate the host password automatically	This essentially means "don't use the host password mechanism" - instead, the host can be unlocked using challenge-response.

-l {(as)|(As)|(AS)}	--lock	Sets the agent lock or system lock.	• "as" Means neither agent nor system lock. A synonym is "none". • "As" Turns on the agent lock but not the system lock • "AS" Turns on both locks. A synonym is "both" • Note that "aS" is disallowed. • The default behavior is "-l none".
-e {(rc)|(Rc)|(RC)}	--enable	Sets the "registration enabled" and "communication enabled" check boxes.	• There are two forms of this. In the first form, just the flags (ex: `-e "RC"`) are specfied; this will apply only to the FS agent. In the second form, the flags are shown applied to each agent just like in the `host show` command. For example: `"fs:RC db2:Rc ids:rc key:rc"` • "rc" means check neither box - registration and communication disabled. A synonym is "none". • "Rc" means check the registration box but not the communication box. • "RC" means both registration and communication boxes are enabled. A synonym is "both" • Note that "rC" is disallowed. • The default is "-e both" (which translates to `"fs:RC db2:rc ids:rc key:rc"`) • Agents not specified default to `rc`. Ex: `-e "ids:RC"` becomes `"fs:rc db2:rc ids:RC key:rc"` • **NOTE**: The group communication settings will override the host communication settings set here.
-s servername	--server	Assign this host to a particular HA server.	
-H <file>	--host-settings	Sets the host's "host settings" to the contents of <file>.	If <file> is "-", read from stdin.

Of the host commands, only the host password must be set. The three options are:

- -G (generate the host password automatically)
- -h (set the host password)
- VMSSC_DEFAULTHOSTPASSWORD set in the environment

Ex: Add host good2.go.com, using all the defaults:

```
vmscc host add good2.go.com
```

Ex: Add a host with a description, and put it in the host group named "rekey_host_group"

```
vmscc host add -d "This is an example description" -g rekey_host_group
good2.go.com
```

Host Modify

```
vmssc host modify [opts] hostname
```

Modify the attributes of an existing host. The word "modify" may be abbreviated "mod". All of the following choices are optional, however at least one must be specified. These options are almost the same as the "add" options:

Argument Description	Long Form Description	Short Description	Notes
-d "		A description of this host.	Don't forget to quote multi-word strings for the shell.
-D	-docker	Flag to enable Docker	(Expects 0 or 1) Once set it cannot be removed
-p port	--port	The port the host uses.	Defaults to 7024.
-g [+/-] hostgroup	--group	Add (or remove) the host to (or from) a hostgroup.	The name of the hostgroup may be prefixed with a "+" (meaning add) or a "-" (meaning remove). If neither are specified, a "+" is implied. Synonym: --hostgroup
-h hostpass	--hostpass	Set the host password.	Synonym: --password
-G	--generate	Generate the host password automatically	This essentially means "don't use the host password mechanism" - instead, the host can be unlocked using challenge-response.
-l {(as)|(As)|(AS)}	--lock	Sets the agent lock or system lock.	• "as" Means neither agent nor system lock. A synonym is "none". • "As" Turns on the agent lock but not the system lock • "AS" Turns on both locks. A synonym is "both" • Note that "aS" is disallowed. • The default behavior is "-l none".

-e {(rc)|(Rc)|(RC)}	--enable	Sets the "registration enabled" and "communication enabled" check boxes.	• There are two forms of this. In the first form, just the flags (ex: `-e "RC"`) are specfied; this will apply only to the FS agent. In the second form, the flags are shown applied to each agent just like in the `host show` command. For example: `"fs:RC db2:Rc ids:rc key:rc"` • "rc" means check neither box - registration and communication disabled. A synonym is "none". • "Rc" means check the registration box and not communication. • "RC" means both registration and communication enabled. A synonym is "both" • Note that "rC" is disallowed. • The default is "-e both" (which translates to only modifying the FS agent to `RC` • Agents not specified aren't altered. Ex: `-e "ids:RC"` will only change the IDS agent to `RC` • **NOTE**: The group communication settings will override the host communication settings set here.
-s servername	--server	Assign this host to a particular HA server.	
-H <file>	--host-settings	Sets the host's "host settings" to the contents of <file>.	If <file> is "-", read from stdin.
-D	--docker	Flag to enable docker. (Expects 0 or 1)	Once set it cannot be removed.
-L license type	--license	The license type to use for this host.	Valid types: term, hourly, perpetual.
-o	--enableldt	Flag to enable LDT (0 or 1)	Used to enable or disable live data transform on a host. Once set it cannot be removed
-c	--conciselog	Flag to enable concise logging.	Expects 0 or 1.

Ex: Remove good2.go.com from one hot group and then put it into another

```
vmssc host mod -g -rekey_host_group
vmssc host mod -g +production_host_group
```

Host Delete

```
vmssc host delete [opts] hostname
```

Delete a host. The word "delete" may be replaced with "del".

There is only one option:

- **-f** Force. Without this flag, the host is not deleteable if it is not reachable by the server. When the **-f** flag is set, the server will delete it regardless.

Ex: delete good2.go.com

```
vmssc host del -f good2.go.com
```

Host Addgp

This adds a guard point to a host. Format: `vmssc host addgp [options] hostname`. Options:

- **-d directory** The full path of the guard point. Also could mean d for "device" for the path to a raw device. Required.
- **-p policy** The policy to apply. Required.
- **-t type** The guard point type. Valid values are "dir", "manualdir", "raw", and "manualraw". If not specified, the default is "dir".
- **-a** If specified, this is an automount guard point. (Default: not automount.)
- **x | -E** flag to add guard point in disabled mode. Guard point is created in enabled mode by default.
- **-S** Specify this flag if trailing '\' or '/' is not required in directory name.
- **-i** Specify docker image ID.
- **-c** Specify docker image Container.
- **-r** Specify that sparse regions be preserved (1) or not (0) during live data transform.

Example:

```
vmssc host addgp -d /full/path/to/gp -p policyname  good2.go.com
```

Example for adding GP to docker image:

```
vmssc host addgp -d /full/path/to/gp -p policyname  -i <valid_image_id>
<docker_enabled_host_name>
```

A valid image ID would like "sha256:bc668af2b27f909aa48aeb12194483bdf165e44b4879c7b89b39c75f48d6fa0f" i.e. <sha256:256_bit_long_hex_value>.

Example for adding GP to docker container:

```
vmssc host addgp -d /full/path/to/gp -p <policy_name>  -i <valid_image_id> -c
<valid_container_id> <docker_enabled_host_name>
```

A valid container ID would like "6d26f81da011b42da54407542c3cb2570bdbae32dbb2dc5f09a92a06ab306934" i.e. <256_bit_long_hex_value>.

To avoid error, one can use values shown in "host docker" command.

Host Modgp

This modifies a guard point on a host. Currently there is only one thing that can be modified, and that is whether the GP is enabled or not. Format: `vmssc host modgp [options] hostname`. Options

- **-d directory** The full path of the guard point. Also could mean d for "device" for the path to a raw device. Required.
- **-p policy** The policy to apply. Required.
- **-e** Enable, or **-x | -E** Disable.
- **-S** Specify this flag if trailing '\' or '/' is not required in directory name.
- **-i** Specify docker image ID.
- **-c** Specify docker image Container.
- **-r** Specify that sparse regions be preserved (1) or not (0) during live data transform

Example: disable the above GP

```
vmssc host modgp -d /full/path/to/gp -p policyname -x  good2.go.com
```

Host Delgp

Removes a guard point from a host. Format: `vmssc host delgp [options] hostname`. Options:

- **-d directory** The full path of the guard point. Also could mean d for "device" for the path to a raw device. Required.
- **-p policy** The policy to apply. Required.
- **-S** Specify this flag if trailing '\' or '/' is not required in directory name.
- **-i** Specify docker image ID.
- **-c** Specify docker image Container.

Both the directory and policy are required because the following scenario could exist: two guardpoints with the same path but different policies, where one is disabled and one is enabled.

Example: remove the above gp

```
vmssc host delgp -d /full/path/to/gp -p policyname  good2.go.com
```

Host Show

```
vmssc host show [hostname]
```

Display information for a host. If given no host name, it will display all known hosts, one per line. If given a host name, it will display information for the host in the following format. This format was developed to make commands both readable by a person and parsable by a machine.

Options:

- **-H host-settings** Show the host settings in addition to all the other information.

Output format:

```
Hostname    : hostname
Description : desc
OS Info     : 'osType, ossubtype' or 'Not yet registered'
Port        : port
Locked      : (no)|(system)|(fs agent)|(system, fs agent)
Comm flags  : fs:RC db2:RC ids:RC key:RC
License     : TERM
Version[a]  : version
Finger[a]   : fingerprint
GP[x]       : (Enabled|Disabled) policy /full/path
Group[x]    : group

Comm Mode   : Two-Way
LDT Enabled : False
Docker      : Disabled
Concise Log : Enabled
```

The **RC|Rc|rc** in the comm flags corresponds to the **-e** option of add/mod above. The **[a]** stands for agent type: fs, db2, ids. Those lines are only printed if an agent of the corresponding type is registered.

An example:

```
Description : This is a description
OS Info     : Linux, RHEL 5
Port        : 7024
Locked      : no
Comm flags  : fs:RC db2:RC ids:RC key:RC
License     : TERM
Version[fs] : 5.0.1.7
Finger[fs]  : DB:C7:19:0E:A1:BB:99:88:E9:34:22:1F:5F:33:7B:35:91:8F:2E:14
GP[1]       : enabled  encrypt_things /full/path/to/gp/one
GP[2]       : disabled other_policy /full/path/to/gp/two
Group[1]    : encrypting_group
Group[2]    : other_group
Comm Mode   : Two-Way
LDT Enabled : False
Docker      : Disabled
Concise Log : Enabled
```

Host Docker

Following is the syntax of the command:

```
vmssc host docker <Host Name>
```

For example:.

```
$vmssc host docker 10.3.1.0
Docker Image [1] :
Image ID       :
sha256:95612a3264fcea256ed7c179d6e4a5dece55e217cff198bbaeb4a7e554f974ca
Image Name     : rhel:latest
Description    : Imported from -
Version        : 1.4.1-dev
```

```
Docker Image [2] :
Image ID       :
sha256:eeb3a076a0bed31a49599e75d6730feb49671fa07fdb26c681e140ba6f95ff54
Image Name     : centos:latest
Description    :
Version        : 1.9.1
Container [1]  Details :
Container ID   :
d66d28fd7ef29e90abe866832b5fafea841692bc92a5a6e7fd026eded0e52c5f
Container Name : evil_mayer
Container [2]  Details :
Container ID   :
98447b98f2f092162d8a4dc38605e843a3ca96b3ebf23c4cce6f3c6683255796
Container Name : grave_turing
```

The above command shows docker image and container related details.

Host showgp

Following is the syntax of the command:

```
vmssc host showgp <Host Name>
```

E.g.

```
$ ./vmssc host showgp 10.3.1.0
GP[0]               : Disabled pol_1 C:\DB2\NODE0000 DOWN
GP[1]               : Enabled pol_1 /home/vormetric/ DOWN


GP[2]               : Enabled Docker_ROOT_All_CK /a/ UP
statuschk_tm        : 10/20/2016 7:20:28
reason              : N/A
guarded             : Guarded
config_state        : guarded
usage               : free
flags               : 0
policy_keyvers      : 0
lock                : 1
guard_time          : 10/20/2016 7:20:28
policy_name         : Docker_ROOT_All_CK
type                : 1
policy_version      : 1
Docker Image ID     :
sha256:354e698f00fdfabb319f1ee9c02c5f881e87c368d4fc48a7463bc3f9c57b05b4
Docker Container ID :
dd97b879d314ae4fe4994bc1cd5c9655aff24d7107d3860b0e6f8ba7d0d1426f


GP[3]               : Enabled Docker_ROOT_All_CK /b/ UP
statuschk_tm        : 10/20/2016 7:35:35
reason              : N/A
```

```
guarded             : Guarded
config_state        : guarded
usage               : free
flags               : 0
policy_keyvers      : 0
lock                : 1
guard_time          : 10/20/2016 7:35:35
policy_name         : Docker_ROOT_All_CK
type                : 1
policy_version      : 1
Docker Image ID     :
sha256:354e698f00fdfabb319f1ee9c02c5f881e87c368d4fc48a7463bc3f9c57b05b4
Docker Container ID :
dd97b879d314ae4fe4994bc1cd5c9655aff24d7107d3860b0e6f8ba7d0d1426f
```

Group Commands

The group commands are similar to those of the host commands. The word "group" may be replaced with synonyms "hostgroup" and "hg". Valid group commands are:

Group Command	Description
add	adds a host group to the server
modify	modifies attributes of an existing host group
delete	deletes a host group from the server
addgp	adds a GuardPoint to an existing host group
modgp	modifies a GuardPoint on an existing host group
delgp	removes a GuardPoint from an existing host group
show	displays information about the host group(s)

Group Add

```
vmssc group add [opts] groupname
```

The host group options are:

Argument	Long Form	Short Description	Notes
-t	--type	Optional flag.	Expects cluster type, valid cluster types are 'Gpfs' or 'Hdfs'. This switch is to be ignored if host group is not a cluster (For nodes).
-d "Description"	--description	A description of this host.	You must enclose multi-word strings in quotes for the shell.
-e {C|c}	--enable	Sets the "communication enabled" check box.	• There are two forms of this. In the first form, just the flags (ex: `-e "C"`) are specfied; this will apply only to the FS agent. In the second form, the flags are shown applied to each agent just like in the `host show` command. For example: `"fs:C db2:c ids:c key:c"` • "C" means set "communication enabled". • "c" disables communication. • The default is "C" (which translates to `"fs:C db2:c ids:c key:c"`) • Agents not specified default to `c`. Ex: `-e "ids:C"` becomes `"fs:c db2:c ids:C key:c"` • **NOTE**: This communication setting overrides that of member hosts.
-h host	--host	Add host to this host group.	

-l {(as)|(As)|(AS)}	--lock	Sets the agent lock or system lock.	• "as" Means neither agent nor system lock. A synonym is "none". • "As" Turns on the agent lock but not the system lock • "AS" Turns on both locks. A synonym is "both" • Note that "aS" is disallowed. • The default behavior is "-l none".

Ex: Add a host group 'groupname' with one guard point:

```
vmssc group add -d "This is an example host group description" -e:C groupname
```

E.g. HDFS cluster can be added as following:

```
$ vmssc group add -t Hdfs test_hdfs
$ vmssc group show test_hdfs
Group Name  : test_hdfs
Description :
Locked      : no
Comm        : fs:C db2:c ids:c key:c kmip:c
Type        : HDFS/Normal
```

GPFS can be added as following:

```
$ vmssc group add -t gpfs test_gpfs
$ vmssc group show test_gpfs
Group Name  : test_gpfs
Description :
Locked      : no
Comm        : fs:C db2:c ids:c key:c kmip:c
Type        : GPFS
```

Group Modify

```
vmssc group modify [opts] groupname
```

The word "modify" may be replaced by "mod" The options here are similar to group add:

Argument	Long Form	Short Description	Notes
-t	--type	Optional flag.	Expects cluster type, valid cluster types are 'Gpfs' or 'Hdfs'. This switch is to be ignored if host group is not a cluster (For nodes).
-d "Description"	--description	A description of this host.	You must enclose multi-word strings in quotes for the shell.

-l {(as)|(As)|(AS)}	--lock	Sets the agent lock or system lock.	• "as" Means neither agent nor system lock. A synonym is "none". • "As" Turns on the agent lock but not the system lock • "AS" Turns on both locks. A synonym is "both" • Note that "aS" is disallowed. • The default behavior is "-l none".
-e {C|c}	--enable	Sets the "communication enabled" check box.	• There are two forms of this. In the first form, just the flags (ex: `-e "C"`) are specfied; this will apply only to the FS agent. In the second form, the flags are shown applied to each agent just like in the `host show` command. For example: `"fs:C db2:c ids:c key:c"` • "C" means set "communication enabled". • "c" disables communication. • The default is "C" • Agents not specified default to `c`. Ex: `-e "ids:C"` will modify just the IDS agent. • **NOTE**: This group communication setting overrides that of member hosts.
-h [+/-] host	--host	Add host to this host group.	The name of the host may be prefixed with a "+" (meaning add) or a "-" (meaning remove). If neither are specified, a "+" is implied.
-p password	--password	Set all host passwords.	Sets the host password to "password" for all hosts in the host group. Note that this can take some time if there are many hosts in the target host group. Also note that this only affects hosts that are in the group at the time this command is issued; hosts added later won't be changed.
-G	--generate	Generate all host passwords	Re-generate all host passwords for all hosts in the host group. This essentially means "don't use the host password mechanism" - instead, the host can be unlocked using challenge-response.

Ex: change the description of a host group

```
vmssc group modify -d "Different description" groupname
```

Group Delete

```
vmssc group delete groupname
```

Deletes a host group. "delete" may be replaced with "del". This command requires no options.

Ex:

```
vmssc group del groupname
```

Group Addgp

This adds a guard point to a host group. Format: `vmssc group addgp [options] groupname`. Options:

- **-d directory** The full path of the guard point. Also could mean d for "device" for the path to a raw device. Required.
- **-p policy** The policy to apply. Required.
- **-t type** The guard point type. Valid values are "dir", "manualdir", "raw", and "manualraw". If not specified, the default is "dir".
- **-a** If specified, this is an automount guard point. (Default: not automount.)
- **-e** Enable, or **-x** Disable. Default: enabled
- **-i** Specify docker image ID.
- **-c** Specify docker container ID.

Example:

```
vmssc group addgp -d /full/path/to/gp -p policyname groupname
```

Example for adding GP to docker image (in host group):

```
vmssc group addgp -d /full/path/to/gp -p policyname  -i <valid_image_id>
<docker_enabled_host_group>
```

A valid image ID would look like "sha256:bc668af2b27f909aa48aeb12194483bdf165e44b4879c7b89b39c75f48d6fa0f" i.e. <sha256:256_bit_long_hex_value>.

Example for adding GP to a docker container (in host group):

```
vmssc host addgp -d /full/path/to/gp -p <policy_name>  -i <valid_image_id> -c
<valid_container_id> <docker_enabled_host_group>
```

A valid container ID would look like "6d26f81da011b42da54407542c3cb2570bdbae32dbb2dc5f09a92a06ab306934" i.e. <256_bit_long_hex_value>.

To avoid error, one can use values shown in "host docker" command. <docker_enabled_host_group> should have atleast one docker enabled host to apply GP.

Group Modgp

This modifies a guard point in a group. Currently there is only one thing that can be modified, and that is whether the GP is enabled or not. Format: `vmssc group modgp [options] groupname`. Options

- **-d directory** The full path of the guard point. Also could mean d for "device" for the path to a raw device. Required.
- **-p policy** The policy to apply. Required.
- **-e** Enable, or **-x** Disable.
- **-i** Specify docker image ID.
- **-c** Specify docker container ID.

Either **-e** or **-x** is required. The "enabled" ability cannot be set upon GP creation (they're enabled by default), so it must be turned on/off here.

Example: disable the above GP

```
vmssc group modgp -d /full/path/to/gp -p policyname -x groupname
```

Group Delgp

Removes a guard point from a host group. Format: `vmssc group delgp [options] hostname`. Options:

- **-d directory** The full path of the guard point. Also could mean d for "device" for the path to a raw device. Required.
- **-p policy** The policy to apply. Required.

Both the directory and policy are both required because the following scenario could exist: two guardpoints with the same path but different policies, and one is disabled and one enabled.

Example: remove the above gp

```
vmssc group delgp -d /full/path/to/gp -p policyname groupname
```

Group Show

```
vmssc group show [groupname]
```

With no groupname, this command will print out all known host groups, one per line.

If a host group is specified, it will display information for the group in the following format. This format was developed so that the commands are both user readable and machine parsable.

```
Description : desc
Locked      : (no)|(system)|(fs agent)|(system, fs agent)
Comm flags  : (RC)|(Rc)|(rc)
GP[x]       : (enabled|disabled) policy /full/path
Member[x]   : host
```

Example:

```
Description : This is a description.
Locked      : no
Comm flags  : RC
GP[1]       : enabled  policy_new /full/path/new
GP[2]       : disabled policy_old /full/path/old
Member[1]   : alpha.beta.com
Member[2]   : gamma.beta.com
```

Policy Commands

Policy commands provide a method to create and modify policies. Policies are kept in XML format.

Policy Command	Description
show	displays name of policies in the system
save	allows the administrator to save a policy
delete	allows the administrator to delete a named policy

Policy Show

Displays the names of all the policies in the system. This is useful when selecting a policy before applying it to a guard point.

```
$ vmssc policy show
Dataxform
Production
UnDataxform
test1
test2
permit_apply_key_3
```

This command can also be used to save a policy to a file name or display it to standard output (stdout).

Argument	Long Form	Short Description	Notes
*-f file	--filename	A file to save policy xml to	"-" sends it to stdout. If unspecified, the xml isn't shown.
*-v version	--version	Show details of a specific policy version.	Expects a numeric value. This switch should be ignored for visualizing details of latest version.

```
$ vmssc policy show -f <filename> test1
Policy Name    : test1
Policy Version : 1
Description    : (none)

$ ./vmssc policy show pol_1
Policy Name    : pol_1
Policy Version : 0
Description    :
Policy Type    : Online
$ vmssc policy show -f <filename> test1
Policy Name    : test1
Policy Version : 1
Description    : (none)
<?xml version="1.0" encoding="UTF-8" standalone="no"?><Policy Version="0">
<Target>
<Users Name="Contact">
</Users>
<Processes Name="Process">
<ProcessSet Name="process0">
</ProcessSet>
</Processes>
<Libs Name="Library">
</Libs>
<ProcessLibs/>
```

```
<Resources Name="Resource">
</Resources>
<Actions Name="Action">
</Actions>
<Times Name="Time">
</Times>
</Target>
<KeyRules KeyCombiningAlg="first_applicable">
</KeyRules>
<NewKeyRules KeyCombiningAlg="first_applicable">
</NewKeyRules>
<SecurityRules PartialMatch="1" NeverDeny="0"
PermitCombiningAlg="first_applicable">
</SecurityRules>
</Policy>
```

The current version details of a policy (default) can be seen as following:

```
$ ./vmssc policy show pol_1_1
Policy Name    : pol_1_1
Policy Version : 1
Description    :
```

The details of a particular version can be seen as following:

```
$ ./vmssc policy show -v 0 pol_1_1
Policy Name    : pol_1_1
Policy Version : 0
Description    :
```

Policy Save

Administrators can save a policy from a file or standard input (stdin). For example, the policy can be piped in from another file or location. To use stdin, specify "-" as the input file.

Argument	Long Form	Short Description	Notes
-d "Description"	--description	A description of this policy.	You must enclose multi-word strings in quotes for the shell.
-f file	--filename	A file to pick the policy xml up from	"-" read from stdin.
-t	--type	Specify policy type. Namely online, offline, ldt.	
-F	--force	Ignore version warnings	If the version in the policy XML doesn't line up with what's in the server, an error will be thrown. This overrides that error. Use this only if you're sure that no one else could be editing the policy.
-a <0|1>	--all	Flag to specify all set.	Expects 0 or 1 as argument.

-i <0|1>	--ignoreset	Flag to ignore set conflict.	Expects 0 or 1 as argument.

```
$ vmssc policy save [-d "description"] [-f <filename>] -t online policyname
```

Policy Delete

Allows an administrator to delete a policy.

```
$ vmssc policy delete policyname
```

Group Commands

The group commands are similar to those of the host commands. The word "group" may be replaced with synonyms "hostgroup" and "hg". Valid group commands are:

Group Command	Description
add	adds a host group to the server
modify	modifies attributes of an existing host group
delete	deletes a host group from the server
addgp	adds a GuardPoint to an existing host group
modgp	modifies a GuardPoint on an existing host group
delgp	removes a GuardPoint from an existing host group
show	displays information about the host group(s)

Group Add

```
vmssc group add [opts] groupname
```

The host group options are:

Argument	Long Form	Short Description	Notes
-t	--type	Optional flag.	Expects cluster type, valid cluster types are 'Gpfs' or 'Hdfs'. This switch is to be ignored if host group is not a cluster (For nodes).
-d "Description"	--description	A description of this host.	You must enclose multi-word strings in quotes for the shell.

-e {C|c}	--enable	Sets the "communication enabled" check box.	• There are two forms of this. In the first form, just the flags (ex: `-e "C"`) are specfied; this will apply only to the FS agent. In the second form, the flags are shown applied to each agent just like in the `host show` command. For example: `"fs:C db2:c ids:c key:c"` • "C" means set "communication enabled". • "c" disables communication. • The default is "C" (which translates to `"fs:C db2:c ids:c key:c"`) • Agents not specified default to `c`. Ex: `-e "ids:C"` becomes `"fs:c db2:c ids:C key:c"` • **NOTE**: This communication setting overrides that of member hosts.
-h host	--host	Add host to this host group.	
-l {(as)|(As)|(AS)}	--lock	Sets the agent lock or system lock.	• "as" Means neither agent nor system lock. A synonym is "none". • "As" Turns on the agent lock but not the system lock • "AS" Turns on both locks. A synonym is "both" • Note that "aS" is disallowed. • The default behavior is "-l none".

Ex: Add a host group 'groupname' with one guard point:

```
vmssc group add -d "This is an example host group description" -e:C groupname
```

E.g. HDFS cluster can be added as following:

```
$ vmssc group add -t Hdfs test_hdfs
$ vmssc group show test_hdfs
Group Name  : test_hdfs
Description :
Locked      : no
Comm        : fs:C db2:c ids:c key:c kmip:c
Type        : HDFS/Normal
```

GPFS can be added as following:

```
$ vmssc group add -t gpfs test_gpfs
$ vmssc group show test_gpfs
Group Name  : test_gpfs
Description :
Locked      : no
Comm        : fs:C db2:c ids:c key:c kmip:c
Type        : GPFS
```

Group Modify

```
vmssc group modify [opts] groupname
```

The word "modify" may be replaced by "mod" The options here are similar to group add:

Argument	Long Form	Short Description	Notes
-t	--type	Optional flag.	Expects cluster type, valid cluster types are 'Gpfs' or 'Hdfs'. This switch is to be ignored if host group is not a cluster (For nodes).
-d "Description"	--description	A description of this host.	You must enclose multi-word strings in quotes for the shell.
-l {(as)|(As)|(AS)}	--lock	Sets the agent lock or system lock.	• "as" Means neither agent nor system lock. A synonym is "none". • "As" Turns on the agent lock but not the system lock • "AS" Turns on both locks. A synonym is "both" • Note that "aS" is disallowed. • The default behavior is "-l none".
-e {C|c}	--enable	Sets the "communication enabled" check box.	• There are two forms of this. In the first form, just the flags (ex: `-e "C"`) are specfied; this will apply only to the FS agent. In the second form, the flags are shown applied to each agent just like in the `host show` command. For example: `"fs:C db2:c ids:c key:c"` • "C" means set "communication enabled". • "c" disables communication. • The default is "C" • Agents not specified default to `c`. Ex: `-e "ids:C"` will modify just the IDS agent. • **NOTE**: This group communication setting overrides that of member hosts.
-h [+/-] host	--host	Add host to this host group.	The name of the host may be prefixed with a "+" (meaning add) or a "-" (meaning remove). If neither are specified, a "+" is implied.
-p password	--password	Set all host passwords.	Sets the host password to "password" for all hosts in the host group. Note that this can take some time if there are many hosts in the target host group. Also note that this only affects hosts that are in the group at the time this command is issued; hosts added later won't be changed.
-G	--generate	Generate all host passwords	Re-generate all host passwords for all hosts in the host group. This essentially means "don't use the host password mechanism" - instead, the host can be unlocked using challenge-response.

Ex: change the description of a host group

```
vmssc group modify -d "Different description" groupname
```

Group Delete

```
vmssc group delete groupname
```

Deletes a host group. "delete" may be replaced with "del". This command requires no options.

Ex:

```
vmssc group del groupname
```

Group Addgp

This adds a guard point to a host group. Format: `vmssc group addgp [options] groupname`. Options:

- **-d directory** The full path of the guard point. Also could mean d for "device" for the path to a raw device. Required.
- **-p policy** The policy to apply. Required.
- **-t type** The guard point type. Valid values are "dir", "manualdir", "raw", and "manualraw". If not specified, the default is "dir".
- **-a** If specified, this is an automount guard point. (Default: not automount.)
- **-e** Enable, or **-x** Disable. Default: enabled
- **-i** Specify docker image ID.
- **-c** Specify docker image Container.

Example:

```
vmssc group addgp -d /full/path/to/gp -p policyname groupname
```

Example for adding GP to docker image (in host group):

```
vmssc group addgp -d /full/path/to/gp -p policyname  -i <valid_image_id>
<docker_enabled_host_group>
```

A valid image ID would look like "sha256:bc668af2b27f909aa48aeb12194483bdf165e44b4879c7b89b39c75f48d6fa0f" i.e. <sha256:256_bit_long_hex_value>.

Example for adding GP to docker container (in host group):

```
vmssc host addgp -d /full/path/to/gp -p <policy_name>  -i <valid_image_id> -c
<valid_container_id> <docker_enabled_host_group>
```

A valid container ID would look like "6d26f81da011b42da54407542c3cb2570bdbae32dbb2dc5f09a92a06ab306934" i.e. <256_bit_long_hex_value>.

To avoid error, one can use values shown in "host docker" command. <docker_enabled_host_group> should have atleast one docker enabled host to apply GP.

Group Modgp

This modifies a guard point in a group. Currently there is only one thing that can be modified, and that is whether the GP is enabled or not. Format: `vmssc group modgp [options] groupname`. Options

- **-d directory** The full path of the guard point. Also could mean d for "device" for the path to a raw device. Required.
- **-p policy** The policy to apply. Required.
- **-e** Enable, or **-x** Disable.
- **-i** Specify docker image ID.
- **-c** Specify docker image Container.

Either **-e** or **-x** is required. The "enabled" ability cannot be set upon GP creation (they're enabled by default), so it must be turned on/off here.

Example: disable the above GP

```
vmssc group modgp -d /full/path/to/gp -p policyname -x groupname
```

Group Delgp

Removes a guard point from a host group. Format: `vmssc group delgp [options] hostname`. Options:

- **-d directory** The full path of the guard point. Also could mean d for "device" for the path to a raw device. Required.
- **-p policy** The policy to apply. Required.

Both the directory and policy are both required because the following scenario could exist: two guardpoints with the same path but different policies, and one is disabled and one enabled.

Example: remove the above gp

```
vmssc group delgp -d /full/path/to/gp -p policyname groupname
```

Group Show

```
vmssc group show [groupname]
```

With no groupname, this command will print out all known host groups, one per line.

If a host group is specified, it will display information for the group in the following format. This format was developed so that the commands are both user readable and machine parsable.

```
Description : desc
Locked      : (no)|(system)|(fs agent)|(system, fs agent)
Comm flags  : (RC)|(Rc)|(rc)
GP[x]       : (enabled|disabled) policy /full/path
Member[x]   : host
```

Example:

```
Description : This is a description.
```

```
Locked      : no
Comm flags  : RC
GP[1]       : enabled  policy_new /full/path/new
GP[2]       : disabled policy_old /full/path/old
Member[1]   : alpha.beta.com
Member[2]   : gamma.beta.com
```

Key Commands

Key commands are used for both agent keys and key vaulting.

Key Command	Description
show	used to display a particular key or list of keys
showversions	used to show versions of a named key
add	allows the administrator to import or create a new key
modify	allows the administrator to modify settings for an existing key
setattr	allows the administrator to set attributes for an existing key
delattr	allows the administrator to remove attributes for an existing key
delete	allows the administrator to remove an existing key
rotate	allows the administrator to rotate a versioned agent key
clone	Clones a versioned agent key

Key Show

Without a name, key show will display a list of the keys available on the system. Key show can be used to display lists of keys, or one key in particular. It accepts the following flags:

Argument	Long Form	Short Description	Notes
-a	--agent	Display agent keys only	
-v	--vaulted	Display vaulted keys only	
-d	--detail	Provide details on each key	
-f	--key-file	Save the symmetric key to this file	Requires a named key. It should be a vaulted key.
-P	--private-key	Save the private key to this file	Requires a named key
-p	--public-key	Save the public key to this file	Requires a named key
-b	--batch	Specify batch size in between 5001 and 65536 (Optional).	Provides option to specify batch size, larger batch size improves response time of command. It can be used for visualizing large number of keys.

Examples:

```
$ ./vmssc show key
agent-host-1clear_key
vaulted-aes-1
vaulted-aes-2
vaulted-aes-3
vaulted-rsa-2048

$ ./vmssc show key -a
agent-host-1
clear_key

$ vmssc key show -d k1
Key Name          : k1
Algorithm         : AES128
Description       :
Creation Time     : 2016-09-25 21:46:08.272
Expiration Time   :
Key Type          : Stored on Server
UUID              : f97e24bf-8a69-3db5-891a-bf839a9491cd
source            : From DSM
Refresh Period    : -1
Key Version       : 0
Key Hash          :
3e70d1ed4a0bd5906799d0196d54f5acd99dda9dfb2214091eaf5e166edb1aee4519f725c893c
0a64a1c20233646c1c5
Key Version Life Span: 0
Attributes[ 0]    : Attribute Index=0
Attributes[ 1]    : Cryptographic Usage Mask=127
Attributes[ 2]    : Object Type=SymmetricKey
Attributes[ 3]    : pdktaALGORITHM=KT_AES128
Attributes[ 4]    : pdktaAPP_SPEC_INFO=(null)
Attributes[ 5]    : pdktaCONTACT_INFO=(null)
Attributes[ 6]    : pdktaEXPIRY_DATE=(null)
Attributes[ 7]    : pdktaKEY_USAGE=ONLINE
Attributes[ 8]    : pdktaUNIQUE=FALSE
Attributes[ 9]    : x-VormCanBePlainText=true
Attributes[10]    : x-VormCanNeverBeExported=true
Attributes[11]    : x-VormCanNeverBePlaintext=false
Attributes[12]    : x-VormCanObjectPersist=true
Attributes[13]    : x-VormID=(null)
```

Key ShowVersions

For a named key, key showversions will display detailed information about the available versions of the named key.

```
$ ./vmssc key show aes256
Key Name          : aes256
Algorithm         : AES256
Description       : M
Creation Time     : 2015-08-26 12:22:00.168
Expiration Time   :
Key Type          : Cached on Host
```

```
UUID             : 02-5
source           : From DSM
Refresh Period   : 30
Key Version      : 5
Key Hash         :
4b7e1b79ed3e7168665c981ab5ce058db5b7772c7f9f5be0417a9f5a476d1ce50706fec29fcc0
148cc42c097e5fcd89c

$ ./vmssc key showversions aes256
Key Name         : aes256
Algorithm        : AES256
Description      : M
Creation Time    : 2015-08-26 12:22:00.168
Expiration Time  :
Key Type         : Cached on Host
Refresh Period   : 30
Key Version      : 5
Key Hash         :
4b7e1b79ed3e7168665c981ab5ce058db5b7772c7f9f5be0417a9f5a476d1ce50706fec29fcc0
148cc42c097e5fcd89c
Key Name         : aes256
Algorithm        : AES256
Description      : M
Creation Time    : 2015-08-26 12:21:08.365
Expiration Time  : 2015-08-26 12:22:00.168
Key Type         : Cached on Host
Refresh Period   : 30
Key Version      : 4
Key Hash         :
289d52765cf169fcc1d370cafffaf033491c8f238387bdef964dba2a80347329bf7f965795919
65165848c093d5ddd9e
Key Name         : aes256
Algorithm        : AES256
Description      : M
Creation Time    : 2015-08-26 12:19:32.575
Expiration Time  : 2015-08-26 12:21:08.365
Key Type         : Cached on Host
Refresh Period   : 30
Key Version      : 3
Key Hash         :
7264eea186d91f15459de9afec55a06eb8a055a86216376716fa589e68f41c0f8502b4c7aa511
cd722c3aa88d01bd4ea
Key Name         : aes256
Algorithm        : AES256
Description      : M
Creation Time    : 2015-08-26 11:11:22.36
Expiration Time  : 2015-08-26 12:19:32.563
Key Type         : Cached on Host
Refresh Period   : 30
Key Version      : 2
Key Hash         :
f81541835050532d15a6657fca704c8c883088df07f041aa4e57a0f79d3c89f5d0a3751588537
374a10b36a79f5a5254
Key Name         : aes256
Algorithm        : AES256
Description      : M
Creation Time    : 2015-08-26 10:44:06.154
```

```
Expiration Time  : 2015-08-26 11:11:22.336
Key Type         : Cached on Host
Refresh Period   : 30
Key Version      : 1
Key Hash         :
9b909de3d7b5637d116806391f02372fffb4ab770f97ba3f0ac462a85164fe2c8e3bb95b8d834
a194196edc628c70066
Key Name         : aes256
Algorithm        : AES256
Description      :
Creation Time    : 2014-10-03 10:03:55.621
Expiration Time  : 2015-08-26 10:44:06.084
Key Type         : Cached on Host
Refresh Period   : 30
Key Version      : 0
```

Key Add

Allows an administrator to import or add by creating a key.

Argument	Long Form	Scope	Short Description	Notes
-d	--description	global	Key description	
-c	--created	global	Creation date of the key.	If not set, the current date will be used. The time-stamp format is yyyy-mm-dd hh:mm:ss[.fffffffff]
-e	--expires	global	Expiration date of the key.	The time-stamp format is yyyy-mm-dd hh:mm:ss[.fffffffff]
-t	--type	global	Key type or algorithm	Valid values are AES128, AES256, ARIA128, ARIA256, 3DES, RSA1024, RSA2048, and RSA4096
-f	--file	global	A file containing the contents of a symmetric key.	If the file is "-", read contents from stdin.
-P	--private-key	global	A file containing the contents of an asymmetric private key	If the file is "-", read contents from stdin.
-p	--public-key	global	A file containing the contents of an asymmetric public key	If the file is "-", read contents from stdin.
-k	--key	global	Symmetric key	Requires 48 hex character key for triple DES, 32 hex characters for AES128 or ARIA128; or 64 hex characters for AES256 or ARIA256.
-V	--create	agent	Create the key (not vaulted)	
-h	--host	agent	Agent key is cached on host	Default behavior. One of -h, -s, or -v is required.
-s	--server	agent	Agent key is stored on server (not cached on host)	One of -h, -s, or -v is required.
-u	--no-unique	agent	Key is non-unique to host	Opposite of -U. Default behavior.
-U	--unique	agent	Key is unique to host	Opposite of -u. Don't use this type of key when guarding shared storage.

-v	--vaulted	vault	Vault the key	One of -h, -s, or -v is required. Vaulted keys are assumed to be third party keys.
-l	--validate	vault	Validate the key before vaulting	Default behavior.
-L	--no-validate	vault	Do not validate the key before vaulting	
-S	--lifespan	agent	Key life in days before automatic rotation	Allows adding a key with a specified lifespan in days, after which the key will be automatically rotated by the DSM.

Examples:

```
$ ./vmssc key add -d "My first key" -e "2015-04-01 01:00:00" -t AES256 -h
first-agent-key

$ ./vmssc key show first-agent-key
Key Name         : first-agent-key
Algorithm        : AES256
Description      : My first key
Creation Time    : 2012-03-29 14:17:19.04
Expiration Time  : 2015-04-01 01:00:00.0
Key Type         : Cached on Host

$ ./vmssc key add -d "My first vaulted key" -t AES256 -f key256.txt -v -l
first-vault-key

$ ./vmssc key show first-vault-key
Key Name         : first-vault-key
Algorithm        : AES256
Description      : My first vaulted key
Creation Time    : 2012-03-29 14:22:39.705
Expiration Time  : (none)
Key Type         : Vaulted

$ ./vmssc key add -d "My first RSA vaulted key" -t RSA2048 -P my.priv.pem -p
my.pub.pem -v first-rsa-vaulted

$ ./vmssc key show first-rsa-vaulted
Key Name         : first-rsa-vaulted
Algorithm        : RSA2048
Description      : My first RSA vaulted key
Creation Time    : 2012-03-29 14:42:48.592
Expiration Time  : (none)
Key Type         : Vaulted


$ ./vmssc key add -d "My first vaulted key" -t AES256 -v -l -k
"5069B0BE3E68E7FFAB2CAE0369B73D267A946D2BCB995BE29FE757F50A37CA05" first-
vault-key-2

$ ./vmssc -u voradmin -p Ssl12345# -s 10.3.27.58 -d domain1 key show first-
vault-key-2
Key Name         : first-vault-key-2
Algorithm        : AES256
```

```
Description      : My first vaulted key
Creation Time    : 2015-08-07 02:35:40.478
Expiration Time  :
Key Type         : Vaulted
UUID             : 02-368973
source           : From DSM
Refresh Period   :
```

Key Modify

Allows an administrator to modify settings for an existing key.

Argument	Long Form	Short Description	Notes
-d	--description	Key description	
-c	--created	Creation date of the key.	If not set, the current date will be used. The time-stamp format is yyyy-mm-dd hh:mm:ss[.fffffffff]
-e	--expires	Expiration date of the key.	The time-stamp format is yyyy-mm-dd hh:mm:ss[.fffffffff]
-h	--host	Agent key is cached on host	Default behavior. One of -h, -s, or -v is required.
-s	--server	Agent key is stored on server (not cached on host)	One of -h, -s, or -v is required.
-S	--lifespan	Key life in days before automatic rotation	Allows modification of the key's lifespan in days, after which the key will be automatically rotated by the DSM.

Example:

```
$ ./vmssc key mod -d "Modified rsa key description" -e "2016-01-02 12:34:56"
first-rsa-vaulted

$ ./vmssc key show first-rsa-vaulted
Key Name         : first-rsa-vaulted
Algorithm        : RSA2048
Description      : Modified rsa key description
Creation Time    : 2012-03-29 14:42:48.592
Expiration Time  : 2016-01-02 12:34:56.0
Key Type         : Vaulted
```

Admin Commands

Admin commands perform actions on administrative accounts in the DSM. For instance, you use these commands to create admins, assign admins to domains, and remove administrative accounts. The terms "user", "administrator", and "admin" are all synonyms; these terms can be used interchangeably. For example, "vmssc admin show" can also be expressed as "vmssc user show".

Admin Command	Description
show	shows the properties of an admin
add	creates an admin account
modify	modifies admin attributes
delete	removes an admin
adddom	adds an admin to a domain
moddom	modifies domain attributes of an admin
deldom	deletes an admin from a domain

The Admin commands require significant knowledge of how domains work in the DSM, and what permissions the different admin types have. A corresponding issue is that in the [authentication options] part of the command, you explicitly state the domain that one operates in, but the domain is implied (from the authentication options) in the [command arguments] part of the command. This results from the nature of actions being performed either inside a particular domain or outside of all domains.

The following is an example of the VMSSC command-line structure:

```
vmssc [auth options] command_group command <command-options-and-arguments>
```

This table shows the tasks that each admin type can perform.

Type	Abbre-viation	Create admin (admin)	Assign To Domain	Server	Audit/ Group /Host /Policy/ Key/ Sig	Domain (Show/ Add /Modify/del)	Cert-ificate	Report	Syslog	Backup	shared secret	Docker host addgp/ delgp	LDT
System	sys	Yes	First only	Yes	No	Yes	No	No	Agent: No	Yes	No	No	No
Domain	dom	No	Yes	Yes	No	No	No	Yes	System: Yes Agent: No	Yes	No	No	No
Security	sec	No	No	Yes	Yes	No	Yes	Yes	System: Yes No	Yes	No	No	No
Domain Security	domsec or ds	No	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
All	all	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

The sub-commands "show", "add", "mod", and "del" perform similar functions to those same commands in the other groups. The sub-commands "adddom", "moddom", and "deldom" are used to add, modify, and remove an admin to or from a domain.

Additionally, any of the above user type account can be a read-only account. Such read-only accounts are not given privilege to execute add/modify/del sub-commands in any command group. However, they have access to 'show' sub-command and any other sub-command except add/modify/del. The key idea behind read-only type users is to provide right to only overview settings/configurations.

Admin Show

Display the properties of an admin. With no arguments, this command displays all the admins in the system. If operating inside a domain, the admins inside that domain are displayed. It has the following options:

Argument	Long Form	Short Description	Notes
-o "Domain"	--resdom	Expects domain name of domain restricted user.	It is optional.

Example:

Outside of a domain:

```
$ ./vmssc admin show
admin
domadmin
foo
glarch
```

Inside a domain:

```
$ ./vmssc -d domain1 admin show
--- Administrators ---
```

```
admin
domadmin
foo
glarch
--- Members of Domain "domain1" ---
domadmin
foo
glarch
```

For a particular admin:

```
$ ./vmssc admin show admin
Name              : admin
Type              : System
Description       : Initial system account
Read-only         : No
```

For local domain (restricted) administrator:

```
$ ./vmssc user show -o local_domain
--- Administrators ---
ld_user
--- Members of Domain "local_domain" ---
ld_user

$ ./vmssc user show -o local_domain ld_user
Name                        : ld_user
Type                        : DomainSecurity
Description                 :
Read-only                   : Yes
Domain local_domain Enabled : Yes
Domain local_domain Roles   : audit, key, policy, host, CR
```

Admin Add

Create an admin account. Arguments:

Argument	Long Form	Short Description	Notes
-d "Description"	--description	A description of this admin.	Don't forget to quote multi-word strings for the shell.
-p "Password"	--password	The initial password for this admin.	The admin must change this password upon first login. Required.
-t "Type"	--type	Type of admin account to create.	The types are specified in the table above. Default: Security
-D	--addtodom --add2dom --domadd	Add the newly created adminto a domain.	The domain is specified in the [authentication options] section of the command, and can come from the command line, vmssc.conf file, or the environment. Note that this command can fail if the admin doing the creating does not have permission to assign additional admins to a domain (as is the case with the System type).

-r "Roles"	--role --roles	The roles to assign to the admin in the domain.	Requires -D. Valid roles are "audit", "host", "key", and "policy". Combinations are possible with a list separated by spaces and commas. Examples: "audit, host"; "audit, host, key, policy". Shortcuts are "all", and each role can be abbreviated to its first letter: "a,h,k,p".
-o "Domain"	--resdom	Expects domain name of domain restricted user.	It is optional.
-R	--readonly	Optional flag.	This flag is to be specified while creating read only user account.

Examples:

```
$ ./vmssc admin add -p P@5sw0rd -d "This user is a newbie" newuser
$ ./vmssc admin show newuser
Name              : newuser
Type              : Security
Description       : This user is a newbie
Read-only         : No

$ ./vmssc -d domain1 admin add -d "All powerful" -t all -p P@5sw0rd -D -r all
master
$ ./vmssc -d domain1 admin show master
Name                   : master
Type                   : All
Description            : All powerful
Read-only              : No
Domain domain1 Enabled : Yes
Domain domain1 Roles   : audit, key, policy, host
```

An administrator can also be added to a domain as following (Create a domain and add first administrator to it.):

```
$ ./vmssc domain add -d "The Final Frontier" -u newuser Space
```

A local domain (restricted) user can be added by a system administrator as following:

```
$ ./vmssc user add -p Helo12345! -o local_domain -t ds ld_user
```

The above command adds a domain/security user by the help of a system administrator credentials.

A read-only account can be created as following:

```
$ ./vmssc user add -p Helo12345! -R -t dom ld_user
```

Admin Modify

This command modifies attributes of an existing admin.

Argument	Long Form	Short Description	Notes

-d "Description"	--description	A description of this admin.	Don't forget to quote multi-word strings for the shell.
-p "Password"	--password	The initial password for this admin.	Resets the admin's password.
-o "Domain"	--resdom	Expects domain name of domain restricted user.	It is optional.
-R	--readonly	Optional flag.	This flag is to be specified for read only account.

Example:

```
$ ./vmssc -d domain1 admin add -d "All powerful" -t all -p P@5sw0rd master

$ ./vmssc -d domain1 admin show master
Name              : master
Type              : All
Description       : All powerful
Read-only         : No
$ ./vmssc admin mod -d "Secret flaw: kryptonite" master
$ ./vmssc admin show master
Name              : master
Type              : All
Description       : Secret flaw: kryptonite
Read-only         : No
$ ./vmssc admin mod -p NewP@5sw0rd master
```

A local domain (restricted) administrator can modify a domain user details as following:

```
$ ./vmssc -u ld_user -p Helo12345! -l local_domain user modify -p Helo12345#
-o local_domain ld_user
```

Following command can be used for modifying a normal user to read-only user:

```
$ ./vmssc user modify -R <user_name>
```

Similarly, a read-only user can be modified back to normal user as following:

```
$ ./vmssc user modify <read_only_user>
```

Admin Delete

Removes the specified admin from the system. It takes the following argument in addition to the admin you want to delete.

Argument	Long Form	Short Description	Notes
-o "Domain"	--resdom	Expects domain name of domain restricted user.	It is optional.

Example:

```
$ ./vmssc admin del
```

```
Expected a username
$ ./vmssc admin del master
[USR0409E] Error from testdsm3.i.vormetric.com: Users "master" cannot be
deleted because they are in domains.
$ ./vmssc -d domain1 admin deldom master
$ ./vmssc admin del master
$ ./vmssc show admin master
[USR0160E] Error from testdsm3.i.vormetric.com: User name "master" does not
exist.
```

A local domain (restricted) user can be deleted as following:

```
$ ./vmssc user del -o local_domain ld_user
[USR0409E] Error from x-dsm: User "ld_user" cannot be deleted because they
are in domains.
$./vmssc -u domain_admin -p Helo12345# -l local_domain user deldom ld_user
$./vmssc user del -o local_domain ld_user
```

Admin Adddom

Adds an admin to a domain. A synonym is "domain adduser".

Argument	Long Form	Short Description	Notes
-r "Roles"	--role --roles	The roles to assign to the admin in the domain.	Valid roles are "audit", "host", "key", and "policy". Combinations are possible with a list separated by spaces and commas. Examples: "audit, host"; "audit, host, key, policy". Shortcuts are "all", and each role can be abbreviated to its first letter: "a,h,k,p".

Admin Moddom

Modifies the domain attributes of an admin already in a domain. A synonym is "domain moduser".

Argument	Long Form	Short Description	Notes
-r "Roles"	--role --roles	The roles to assign to the admin in the domain.	Valid roles are "audit", "host", "key", and "policy". Combinations are possible with a list separated by spaces and commas. Examples: "audit, host"; "audit, host, key, policy". Shortcuts are "all", and each role can be abbreviated to its first letter: "a,h,k,p".
-e	--enable	Enable an admin in the domain	Obviously a counterpart to -x below.
-x	--disable	Disable an admin in the domain	The opposite of -e.

Example:

```
$ ./vmssc user moddom -x newuser
```

Admin Deldom

Remove an admin from a domain. A synonym is "domain deluser". This command takes no arguments.

E.g. adddom, moddom, and deldom commands can be used as following:

```
$ ./vmssc admin add -p P@sSw0rd -d "newb" newuser

$ ./vmssc -d Earth admin adddom -r policy,key newuser

$ ./vmssc -d Earth admin show newuser
Name                 : newuser
Type                 : Security
Description          : newb
Read-only            : No
Domain Earth Enabled : Yes
Domain Earth Roles   : key, policy

$ ./vmssc -d Earth admin moddom -r all -x newuser

$ ./vmssc -d Earth admin show newuser
Name                 : newuser
Type                 : Security
Description          : newb
Read-only            : No
Domain Earth Enabled : No
Domain Earth Roles   : audit, key, policy, host

$ ./vmssc -d Earth admin deldom newuser

$ ./vmssc -d Earth admin show newuser
Name              : newuser
Type              : Security
Description       : newb
Read-only         : No
```

Key Setattr

Allows an administrator to set attributes for an existing key.

Argument	Long Form	Short Description	Notes
-a	--attribute	An attribute as a "name=value" pair.	Spaces are allowed in the name component
-A	--attribute-file	A file containing "name=value" pairs.	Spaces are not allowed in the name component

Example:

```
$ ./vmssc key setattr -a "Usage=Experimentation" first-rsa-vaulted

$ cat attrs.txt
Organization = Development
Purpose = Example
FavoriteColor = Blue

$ ./vmssc key setattr -A attrs.txt first-rsa-vaulted

$ ./vmssc key show first-rsa-vaulted
Key Name          : first-rsa-vaulted
Algorithm         : RSA2048
Description       : Modified rsa key description
Creation Time     : 2012-03-29 14:42:48.592
Expiration Time   : 2016-01-02 12:34:56.0
Key Type          : Vaulted
Attributes[ 0]    : FavoriteColor=Blue
Attributes[ 1]    : Organization=Development
Attributes[ 2]    : Purpose=Example
Attributes[ 3]    : Usage=Experimentation
```

Key Delattr

Allows an administrator to delete an attribute from an existing key.

Argument	Long Form	Short Description	Notes
-a	--attribute	Name of the attribute to be delete	

Example:

```
$ ./vmssc key delattr -a "Purpose" first-rsa-vaulted

$ ./vmssc key show first-rsa-vaulted
Key Name          : first-rsa-vaulted
Algorithm         : RSA2048
Description       : Modified rsa key description
Creation Time     : 2012-03-29 14:42:48.592
Expiration Time   : 2016-01-02 12:34:56.0
```

```
Key Type          : Vaulted
Attributes[ 0]    : FavoriteColor=Blue
Attributes[ 1]    : Organization=Development
Attributes[ 2]    : Usage=Experimentation
```

Key Delete

Allows an administrator to delete an existing key.

Example:

```
$ ./vmssc key delete first-rsa-vaulted
```

Key Rotate

Allows an administrator to rotate an existing versioned agent key.

Argument	Long Form	Short Description	Notes
-r	--reason	Reason code	One of "M" for maintenance, "C" for compromise, or "O" for other. If "O" is selected, a description is also required.
-d	--description	Detailed description	Optional description of the rotation. This will be combined with the reason code to provide the description text available for the new version of the key, e.g. "O: annual key rotation". Mandatory for reason code "O"

Examples:

```
$ ./vmssc key rotate -r M agentkey1
$ ./vmssc key rotate -r O -d "Annual rotation required" agentkey2
```

Key Clone

Allows an administrator to clone an existing versioned agent key.

Argument	Long Form	Notes
-n	--newkey	Expects new key name, mandatory option.
-d	--description	Optional description of the key.
-t	--type	Type of new key, valid types are sos (Stored on Server), coh (Cached on Host) and vault (key vault).
-C	--creation	Creation date of key.
-E	--expiration	Expiration date of the key.

-c	--cached	Key's time to live.
-l	--life	Key version life span.
-u	--uniquetohost	Flag, If set key value is different on each host.
-v	--version	Key version to clone, mandatory option.

Examples:

```
$ ./vmssc key clone -v 0 -n intial_key_cloned initial_key
```

Domain Commands

Administrative Domains, referred to as "Domains" in this document, can be thought of as "silos" where administrators, keys, policies, and hosts reside. The Domain commands are used to add, show, modify, and remove domains.

Domain Command	Description
show	shows available domains and domain properties
add	adds a new domain to the server
modify	modifies an existing domain
quota	adjusts the per-domain license quotas
delete	removes or deletes the specified domain

Domain Show

Like the other "show" commands, if no arguments are given all the domains are shown; if the name of a domain is given then the properties of that domain are shown. Examples:

```
$ ./vmssc domain show
domain1
domain2
Earth

$ ./vmssc domain show Earth
Name           : Earth
Description    : The whole world
Helpdesk       : Call the global support desk
License Header : Agt  Type    Quota  Usage  Expiry
License[0]     : DB2  TERM    1      0      Sep-30-2012
License[1]     : IDS  TERM    -      0      Sep-30-2012
License[2]     : FS   TERM    -      0      Sep-30-2012
License[3]     : KEY  TERM    -      2      Sep-30-2012
License[4]     : DB2  HOURLY  1      0      -
License[5]     : IDS  HOURLY  0      0      -
License[6]     : FS   HOURLY  -      0      -
License[7]     : KEY  HOURLY  -      0      -
```

Domain Add

Add a domain to the DSM. Arguments for adding a domain are as follows:

Argument	Long Form	Short Description		Notes
-d description	--description	Set the description for this domain	Optional	
-h helpdesk	--helpdesk	Sets the helpdesk text that's displayed on the agent.	Optional	

-u user	--user	The first administrator for this domain	Optional. Once one is specified, another cannot be added.
-o organization	--organization	Expects organization name.	Optional.

'domain add' command allows user to create a domain as well as assign first administrator to it as following:

```
$ ./vmssc domain add -d "The Final Frontier" -u voradmin Space

$ ./vmssc domain show Space
Name        : Space
Description : The Final Frontier
Helpdesk    :
```

Domain Modify

The arguments to modify a domain are as follows:

Argument	Long Form	Short Description	Notes
-d description	--description	Set the description for this domain	Optional
-h helpdesk	--helpdesk	Sets the helpdesk text that's displayed on the agent.	Optional
-u user	--user	The first administrator for this domain	Optional. Once one is specified, another cannot be added.

They are the same as those for adding a domain. For example:

```
$ ./vmssc domain mod -h "These are the voyages into a new frontier" Space

$ ./vmssc show domain Space
Name        : Space
Description : The Final Frontier
Helpdesk    : These are the voyages into a new frontier
```

Domain Quota

This call adjusts the per-domain license quota. This quota was displayed previously as output of the domain show command. This call will adjust one row of the table in domain show.

Argument	Long Form	Short Description	Notes
-a agent type	--agent	Specifies the agent type to modify. Valid agents are fs, db2, ids, and key.	Required
-t license type	--type	Specifies the license type to modify. Valid types are term, hourly, and perpetual.	Required
-q quota	--quota	Set to this quota.	"None" clears.

-e expiry date	--expiry	Set the expiry date.	Term licenses only. "None" clears.

Examples:

```
$ ./vmssc domain quota -a fs -t term -q 10 Earth
$ ./vmssc domain show Earth
Name           : Earth
Description    : The whold world
Helpdesk       : Call the global support desk
License Header : Agt  Type    Quota  Usage  Expiry
License[0]     : DB2  TERM    1      0      Sep-30-2012
License[1]     : IDS  TERM    -      0      Sep-30-2012
License[2]     : FS   TERM    10     0      Sep-30-2012
License[3]     : KEY  TERM    -      2      Sep-30-2012
License[4]     : DB2  HOURLY  1      0      -
License[5]     : IDS  HOURLY  0      0      -
License[6]     : FS   HOURLY  -      0      -
License[7]     : KEY  HOURLY  -      0      -

$ ./vmssc domain quota -a fs -t term -e Sep-19-2012 Earth
$ ./vmssc domain show Earth
Name           : Earth
Description    : The whold world
Helpdesk       : Call the global support desk
License Header : Agt  Type    Quota  Usage  Expiry
License[0]     : DB2  TERM    1      0      Sep-30-2012
License[1]     : IDS  TERM    -      0      Sep-30-2012
License[2]     : FS   TERM    -      0      Sep-19-2012
License[3]     : KEY  TERM    -      2      Sep-30-2012
License[4]     : DB2  HOURLY  1      0      -
License[5]     : IDS  HOURLY  0      0      -
License[6]     : FS   HOURLY  -      0      -
License[7]     : KEY  HOURLY  -      0      -
```

Domain Delete

Removes a domain. Only one argument is accepted; the name of the domain. Example:

```
$ ./vmssc domain del Space
```

Certificate Commands

The Certificate commands add, show, search and delete certificates. You can also store and report on certificates.

Certificate Command	Description
add	adds a new certificate to the server
show	downloads and/or displays a certificate
search	similar to show, but filters or refines a search
delete	deletes a named certificate from the server

Cert Add

Upload one certificate. A unique identifier for that certificate is returned.

```
cert add [options] <cert-file>
```

Argument	Long Form	Short Description	Notes
-f type	--format	Specify the format of the certificate file	If not specified, chooses based on the file extension. Valid types are "der", "pem", "pkcs7", or "pkcs12"
-P filename	--priv	Private key	The specified file contains the private key associated with this certificate.
-p password	--password	Password for .p12/.pfx files	Need this to open this type of file and analyze the contents
-d description	--desc	description	

Common usage:

```
$ vmssc cert add mycert.pem
Newly imported certificate has ID 1067

$ vmssc cert add mycert.pem -d 'test description'
Newly imported certificate has ID 1067
```

Cert Show

Display and/or download one certificate.

```
cert show [options] <id>
```

Argument	Long Form	Short Description	Notes
-f file	--file	Output into file specified by <file>	This will come back in the format that it was originally added.

-P filename	--priv	Private key	Write the private key associated with this certificate into specified file.
-p password	--password	Password for .p12/.pfx files	This is an option so that it's only displayed when you want it

Common usage:

```
$ vmssc cert show 1067
ID          : 1067
Subject     : CN=ORname_Jungo: OpenRG Products Group, C=US
Issuer      : CN=ORname_Jungo: OpenRG Products Group, C=US
Issuer ID   : 1067
Key Algo    : RSA
Key Bits    : 1024
Sig Algo    : MD5withRSA
Not Before  : 2004-06-03 12:11:43.0
Not After   : 2024-05-29 12:11:43.0
Lineage     : Root
Usages      : Client Authentication; Code Signing; Email; Server
Authentication;
Is CA?      : True
Alt Names   :
CRL         :
OCSP        :
Orig Format: PEM
Description: test description
```

Cert Search

This command searches on the certificate's attributes. Instead of overloading "show" with "show all objects", the certificate "search" command allows one to specify search criteria and get back a list of applicable certs.

```
cert search [options]
```

Argument	Long Form	Short Description	Notes
-a <algo>	--algo	Public key algorithm	Usually RSA
-A <algo>	--sig-algo	Signature algorithm is <algo>	Substring searching allowed
-b <min bits>	--min-bits	Minimum size of public key algorithm bits	Generally they are 1024/2048
-B <max bits>	--max-bits	Maximum size of public key algorithm bits	Generally they are 1024/2048
-f <text>	--format	The format of the file that was used to store the cert during its original import	Valid types are "der", "pem", "pkcs7", or "pkcs12"
-i <id>	--issuer	The ID of the issuer certificate	
-l <text>	--lineage	Is the certificate a root, intermediate or leaf in a certificate chain?	

-p	--without-privkey	Opposite of -P.	Shows all entry, including certificates with/without private key associated with them.
-P	--with-privkey	There is an associated private key	
-s <text>	--subject	Subject contains <text>	Substring searching allowed
-u <text>	--usages	Does the certificate support usages?	Such as "serverauth", "clientauth", "email", "codesigning", "timestamp", "ocspsign"
-F <from-date>	--from	Date range beginning	Show certificates that begin after this date
-T <to-date>	--to	Date range end	Show certificates that end before this date
-d <desc>	--desc	Description	Show certificates with specified description

Common Usage

```
$ ./vmssc cert search
      1000 : CN=PAULVM-CA, CN=PAULVM-CA, O=vm, C=us
      1001 : OU=Class 3 Public Primary Certification Authority,
O="VeriSign,...
      1002 : CN=Thawte SGC CA, O=Thawte Consulting (Pty) Ltd., C=ZA
      1003 : CN=mail.google.com, O=Google Inc, L=Mountain View,
ST=Californi...
      1004 : CN=mail.futuretrends.us, OU=Domain Control Validated,
O=mail.fu...
      1021 : SERIALNUMBER=07969287, CN=Go Daddy Secure Certification
Authori...
      1060 : CN=remote.epmmgmt.com

$ ./vmssc cert search -b 2048 -B 2048 -F "2010-01-01 00:00:00.0"
      1004 : CN=mail.futuretrends.us, OU=Domain Control Validated,
O=mail.fu...

$ ./vmssc cert search -d 'test description'
      1000 : CN=PAULVM-CA, CN=PAULVM-CA, O=vm, C=us
      Desc : test description
```

Cert Del

Delete a particular certificate.

```
cert del [options] <id>
```

Argument	Long Form	Short Description	Notes
-c	--issued-certs	Also delete all certs that this entity has issued	Only works for certs that were imported in the same file
-a	--issuer-chain	Also delete certificates up to the root	Only works for certs that were imported in the same file

Signature Commands

Valid Signature commands are:

Host Command	Description
add	create a new signature set on server
modify	modifies attributes of an existing signature set
delete	deletes a signature set from the server
show	displays signature information
addsig	adds signature to existing signature set
delsig	deletes signature from existing signature set
sign	sign a signature set
status	get status of signature set signing action
stop	stops signing a signature set

Signature Add

```
vmssc sig add [opts] [signature set name]
```

This command adds a signature set.

Argument	Long Form	Short Description	Notes
-d "Description"	--description	A description of this signature set.	Don't forget to quote multi-word strings for the shell.
-i	--image	Specify docker image ID.	
-c	--container	Specify docker container ID.	

Ex: Create a new signature set named sigSet

```
vmssc sig add sigSet
```

```
vmssc sig add -d "This is a new signature set" sigSet
```

Signature Modify

```
vmssc sig mod [opts] [signature set name]
```

Add/Modifies the description attribute of an existing signature set.

Argument	Long Form	Short Description	Notes
-c	--container	Specify docker container ID.	
-i	--image	Specify docker image ID.	
-d "Description"	--description	A description of this signature set.	Don't forget to quote multi-word strings for the shell.

Ex: Modify description of signature set named sigSet

```
vmssc sig mod -d "This is a new description" sigSet
```

Signature Delete

```
vmssc sig del [signature set name]
```

Delete a Signature set.

Ex: delete signature set 'sigSet'

```
vmssc sig del sigSet
```

Signature Show

```
vmssc sig show [signature set name]
```

Display information of a signature set. If given no signature name, it will display all known signature set, one per line. If given a signature set name, it will display information about that signature set.

It has the following options:

Argument	Long Form	Short Description	Notes
-s	--sigs	Doesn't expect any argument.	Display the list of individual signatures inside this signature set.

Example:

```
$ ./vmssc sig show
sigSetName1
sigSetName2

$ ./vmssc sig show sigSetName1
Name              : Name of the signature set
Status            : Current status of signing in words (e.g.
UNSIGNED,STARTED,IN PROGRESS,ABORTED,FINISHED,FINISHED WITH WARNING etc).
```

```
Percent Complete : If ongoing, the % complete for the signing process

$ ./vmssc sig show -s sigSetName1
fileName1 : signature1
fileName2 : signature2
...
```

Add signature to signature set

```
vmssc sig addsig [opts] [signature set name]
```

This command add signature to signature set. Only one signature can be added at a time.

Options

Argument	Long Form	Short Description	Notes
-f "file name"	--file	A file (with complete path) containing list if file names and their signatures. File name and signature will be separated by space character. Two entries in file will be separated by new line character.	If the file contains invalid signature it will be skipped and no error message will be shown.

Example:

```
$ ./vmssc sig addsig -f /home/xyzuser/dir/sample_sig.txt sigSetName

content of sample_sig.txt
/bin/file1 64_character_long_signature_of_file1
/bin/file2 64_character_long_signature_of_file2
....
```

Delete a signature from signature set

```
vmssc sig delsig [opts] [signature set name]
```

This command removes signature from specified signature set.

Options

Argument	Long Form	Short Description	Notes
-f "file name"	--file	A file (with complete path) containing list of signatures to be removed. Each signature value must be of 64 characters in length. Signature values should be separated by new line character.	If signature contains more than 64 characters, it will be ignored.

Example:

```
$ ./vmssc sig delsig -f /home/xyzuser/dir/sample_sig.txt sigSetName
```

```
content of sample_sig.txt
64_character_long_signature_of_file_to_be_deleted
64_character_long_signature_of_file_to_be_deleted
....
```

Sign a signature set

```
vmssc sig sign [opts] [signature set name]
```

This command signs a set of files on specified host. Generated signatures will be associated to specified signature set.

Options

Argument	Long Form	Short Description	Notes
-h "hostname"	--host	Host on which files to be signed are located.	This is a required option.
-f "source file name"	--file	File (with path) containing list of files to be signed.	Each file/directory name to be signed will be separated by new line character.
-c	--container	Specify docker container ID.	
-i	--image	Specify docker image ID.	
-n "file names"	--name	List of files to be signed.	Entire list should be enclosed in single quotes. Multiple file names are separated by space character. Individual file name should not contain space character.

Example:

```
$ ./vmssc sig sign -h host.i.vormetric.com -f /home/test/source_file.txt
sigSetName

Content of source_file.txt
/some_dir/file1
/some_dir/file2
/file3
/file4
....


$ ./vmssc sig sign -h host.i.vormetric.com -n "/bin/file1 /bin/file2"
sigSetName
```

Get status of ongoing signing action

```
vmssc sig status [opts] [signature set name]
```

This command returns status of ongoing signing of signature set.

Options

Argument	Long Form	Short Description	Notes
-h "hostname"	--host	Host on which files to be signed are located.	This is a required option.

Example:

```
$ ./vmssc sig status -h host.i.vormetric.com sigSetName
```

Stop ongoing signing

```
vmssc sig stop [opts] [signature set name]
```

This command stops ongoing signing of signature set.

Options

Argument	Long Form	Short Description	Notes
-h "hostname"	--host	Host on which files to be signed are located.	This is a required option.

Example:

```
$ ./vmssc sig stop -h host.i.vormetric.com sigSetName
```

Report Commands

The report commands include the following:

Name	Description
submit	Submits request to generate a new report
status	Queries status of a submitted request
cancel	Cancels a submitted request

The report commands provide the ability to request a report on the status of all guard points and hosts per domain. They also provide option to query the status of an already submitted request and ability to cancel a request in progress as shown in the table above.

Following sub-sections describe the report commands, their usage, and examples in detail.

Report Submit

The "submit" command submits a request to generate report on status of all guard-points and hosts. It expects report type to be passed as an argument compulsorily. A valid report type is "HOST_GUARD_POINT_STATUS". Following is the syntax of this command:

```
$ ./vmssc report submit <Report Type>
```

This command assumes that authentication credentials with corresponding domain name is already provided. Domain name is a mandatory field to execute this command.

As example, a sample command to submit a request and corresponding output will be the following:

```
$ ./vmssc report submit HOST_GUARD_POINT_STATUS
Report ID 1011 submitted for generation
```

As in the output shown above, report ID is unique identifier for request submitted.

Report Status

The "status" command queries status of an already submitted request. It expects unique identifier <Report ID> of the report to be passed compulsorily. It is the same ID which is returned when a request is submitted. Following is the syntax of this command:

```
$ ./vmssc report status <Report ID>
```

Status of a report can be either of QUEUED, STARTED, COMPLETED, CANCELLED or ERROR. A sample command to query the status of a request will be:

```
$ ./vmssc report status 1800
Report Type               : HOST_GUARD_POINT_STATUS
```

```
Report ID                 : 1800
Report Status             : COMPLETED
Host Count                : 1
Number of Hosts Completed : 0
Creation Time             : 2015-04-14 16:04:22
Last Update Time          : 2015-04-14 16:05:12
Completion Time           : 2015-04-14 16:05:12
```

The output of command shown above has following data fields. However, some fields such as tasks count, remaining tasks, queue position of request, and completion time are optional and these fields are shown based on state of the report.

1. Data about report
    1. Report identifier
    2. Report name
    3. Report status (QUEUED, STARTED, COMPLETED, CANCELLED, or ERROR).
    4. Current position of request in the queue.
2. Data about hosts
    1. total number of hosts
    2. Hosts completed
    3. total number of sync-status tasks
    4. total number of remaining tasks.
3. Date time information about report
    1. created time
    2. completed time
    3. last update time

Report Cancel

The "cancel" command cancels an already submitted request, and stops an ongoing report generation. It expects Report ID to be passed as an argument. Following is the syntax of this command:

```
$ ./vmssc report cancel <Report ID>
```

A typical command to cancel an ongoing report generation will be:

```
$ ./vmssc report cancel 1052
Report ID 1052 cancelled
```

Rekey Commands

The rekey commands include the following:

Name	Description
resume	Resume a live data transform
pause	Pause a live data transform
getqos	Fetch 'quality of service' (QOS) parameters for live data transform
setqos	Set QOS parameters for live data transform
modifyqos	Modify QOS parameters for live data transform
listsched	List available QOS schedules
getsched	Fetch a named QOS schedule
addsched	Add a new named QOS schedule
updatesched	Update an existing named QOS schedule
deletesched	Delete and existing named QOS schedule

The rekey commands provide the capabilities to examine and modify the settings used to control live data transform for individual VTE hosts, or host groups.

Following sub-sections describe the rekey commands, their usage, and examples in detail.

Rekey Resume

The "resume" command is used to request that the named host should resume any paused live data transform operations.

```
$ ./vmssc rekey resume <agent host name>
```

Rekey Pause

The "pause" command is used to request that the named host should pause any ongoing live data transform operations.

```
$ ./vmssc rekey pause <agent host name>
```

Rekey Get QOS

The "getqos" command is used to display the QOS parameters for a named host or host group. It accepts the following optional argument:

Argument	Long Form	Short Description	Notes
-g	--group	Selects a named host group operation.	If not set, the name parameter refers to a host rather than a host group.

```
$ ./vmssc rekey getqos centos-7-0
QoS CPU Percentage: 100
QoS CPU Cap       : Disabled
QoS Schedule Name : <not set>
```

Rekey Set QOS

The "setqos" command is used to set the QOS parameters for a named host or host group. Any arguments not provided will be set to the default values (see also modifyqos) It accepts the following arguments:

Argument	Long Form	Short Description	Notes
-c	--cap	Set (1 or Y) or unset (0 or N) CPU cap	When cap is set, live data transform will attempt to keep CPU usage below the 'percent' allowed (see below) even on an idle system. When unset, CPU usage could go higher but not at the expense of other processing requirements. If not provided, uncapped is assumed.
-g	--group	Selects a named host group operation.	If not set, the name parameter refers to a host rather than a host group.
-p	--percent	Sets the CPU usage value	Value cannot be more than 100. If not provided, 100% is assumed. A value of 0 is equivalent to 100% - indicating QOS should not monitor rekey CPU utilization and as such rekey CPU utilization may suppress application CPU usage.
-s	--schedule	Selects a named schedule to run live dataxform	If not provided, the current schedule will be cleared for this host or host group.

```
$ ./vmssc rekey setqos -c 1 -p 80 -s MySchedule centos-7-0

$ ./vmssc rekey getqos centos-7-0
QoS CPU Percentage: 80
QoS CPU Cap       : Enabled
QoS Schedule Name : MySchedule
```

Rekey Modify QOS

The "modifyqos" command combines the two operations "getqos" and "setqos", and is used to modify one or more QOS parameters for a named host or host group, while keeping all other parameters unchanged. It accepts the following arguments:

Argument	Long Form	Short Description	Notes

-c	--cap	Set (1 or Y) or unset (0 or N) CPU cap	When cap is set, live data transform will attempt to keep CPU usage below the 'percent' allowed (see below) even on an idle system. When unset, CPU usage could go higher but not at the expense of other processing requirements.
-g	--group	Selects a named host group operation.	If not set, the name parameter refers to a host rather than a host group.
-p	--percent	Sets the CPU usage value	Value cannot be more than 100. If not provided, 100% is assumed. A value of 0 is equivalent to 100% - indicating QOS should not monitor rekey CPU utilization and as such rekey CPU utilization may suppress application CPU usage.
-s	--schedule	Selects a named schedule to run live dataxform	If not provided, the current schedule will be cleared for this host or host group.

```
$ ./vmssc rekey getqos centos-7-0
QoS CPU Percentage: 100
QoS CPU Cap       : Enabled
QoS Schedule Name : MySchedule

$ ./vmssc rekey modifyqos -p 80 centos-7-0

$ ./vmssc rekey getqos centos-7-0
QoS CPU Percentage: 80
QoS CPU Cap       : Enabled
QoS Schedule Name : MySchedule
```

Rekey List Schedule

The "listsched" command is used to list the named QOS schedules. It accepts the following arguments:

Argument	Long Form	Short Description	Notes
-v	--verbose	Enable verbose mode	In verbose mode, full details of the schedules are output instead of just the names and descrptions

```
$ ./vmssc rekey listsched
Name: ANY_TIME Description: Any Day and Any Time of the week
Name: WEEKENDS Description: Weekend Qos Schedule
Name: WEEKNIGHTS Description: Weeknight Qos Schedule

$ ./vmssc rekey listsched -v
Name: ANY_TIME
Desc: Any Day and Any Time of the week
 Sun: 0000-2359
 Mon: 0000-2359
 Tue: 0000-2359
 Wed: 0000-2359
 Thu: 0000-2359
 Fri: 0000-2359
 Sat: 0000-2359
```

```
Name: WEEKENDS
Desc: Weekend Qos Schedule
 Sun: 0000-2359
 Mon: 0000-0700
 Fri: 2100-2359
 Sat: 0000-2359
Name: WEEKNIGHTS
Desc: Weeknight Qos Schedule
 Mon: 0000-0700 2100-2359
 Tue: 0000-0700 2100-2359
 Wed: 0000-0700 2100-2359
 Thu: 0000-0700 2100-2359
 Fri: 0000-0700 2100-2359
```

Rekey Get Schedule

The "getsched" command is used to fetch full details of a named QOS schedule. It has no optional arguments. The output is an a form that can be saved and edited to create a new named schedule.

```
$ ./vmssc rekey getsched weeknights
Name: WEEKNIGHTS
Desc: Weeknight Qos Schedule
 Mon: 0000-0700 2100-2359
 Tue: 0000-0700 2100-2359
 Wed: 0000-0700 2100-2359
 Thu: 0000-0700 2100-2359
 Fri: 0000-0700 2100-2359
```

Rekey Add Schedule

The "addsched" command is used to create a new named QOS schedule. It accepts the following arguments:

Argument	Long Form	Short Description	Notes
-d	--desc	Description of schedule	The description should be used to provide additional detail on how the schedule is intended to be used. This parameter is required if no "Desc:" line is present in the schedule file. If both are present, the command line switch takes precedence.
-f	--filename	Name of input file containing the schedule parameters.	A - may be used to indicate input from standard input.

The name of the schedule may be provided using the "Name:" entry in the schedule file, or on the command line. If both are provided, they must match.

```
$ ./vmssc rekey addsched -f newschedule

$ echo "Mon: 0000-0559" | ./vmssc rekey addsched -f - -d "Monday Morning
only" MonMorn

$ ./vmssc rekey listsched
Name: ANY_TIME Description: Any Day and Any Time of the week
Name: MonMorn Description: Monday Morning only
Name: NewSchedule Description: Weekend and week nights schedule
Name: WEEKENDS Description: Weekend Qos Schedule
Name: WEEKNIGHTS Description: Weeknight Qos Schedule
```

Rekey Update Schedule

The "updatesched" command is used to changed the parameters of an existing named QOS schedule. It accepts the following arguments:

Argument	Long Form	Short Description	Notes
-d	--desc	Description of schedule	The description should be used to provide additional detail on how the schedule is intended to be used. This parameter is required if no "Desc:" line is present in the schedule file. If both are present, the command line switch takes precedence.
-f	--filename	Name of input file containing the schedule parameters.	A - may be used to indicate input from standard input.

The name of the schedule may be provided using the "Name:" entry in the schedule file, or on the command line. If both are provided, they must match.

```
$ ./vmssc rekey getsched MonMorn
Name: MonMorn
Desc: Monday Monring only
 Mon: 0000-0559

$ ./vmssc rekey getsched MonMorn | ./vmssc rekey updatesched -f - -d "Monday
Morning Only"

$ ./vmssc rekey getsched MonMorn
Name: MonMorn
Desc: Monday Morning Only
 Mon: 0000-0559
```

Rekey Delete Schedule

The "deletesched" command is used to delete a named QOS schedule. The built-in predefined schedules provided by the DSM may not be deleted.

```
$ ./vmssc rekey listsched
Name: ANY_TIME Description: Any Day and Any Time of the week
Name: MonMorn Description: Monday Morning Only
Name: NewSchedule Description: Updated
Name: WEEKENDS Description: Weekend Qos Schedule
Name: WEEKNIGHTS Description: Weeknight Qos Schedule

$ ./vmssc rekey deletesched MonMorn

$ ./vmssc rekey listsched
Name: ANY_TIME Description: Any Day and Any Time of the week
Name: NewSchedule Description: Updated
Name: WEEKENDS Description: Weekend Qos Schedule
Name: WEEKNIGHTS Description: Weeknight Qos Schedule

$ ./vmssc rekey deletesched WEEKENDS
[DAO01347E] Error from dsm2: System defined ODT Schedule WEEKENDS in mydomain
cannot be deleted
```

Syslog Commands

The syslog commands include the following:

Name	Description
addhost	Adds syslog server for an agent.
showhost	Shows syslog servers associated with an agent.
add	Adds syslog host settings to DSM.
delete	Deletes a syslog host from DSM.
show	Shows syslog host setting from DSM.

The syslog commands provide the ability to perform various operations such as adding, deleting, visualizing and configuring syslog servers. Some of these commands are restricted and can be used by a system administrator only. These commands are domain specific hence a syslog server can be configured in a domain or at the system level of the DSM. Apart from these, a syslog server can also be configured for logging events of an agent.

Syslog Addhost

The "addhost" command allows a user to configure and add a syslog server for an agent. These are the following options of this command:

Argument	Long Form	Short Description	Notes
-a	--agenttype	Expects agent type as an argument.	Agent type is a mandatory field, a valid agent type is FS.
-h	--host	Expects a host name.	Host name is an optional field.
-o	--server1	Expects host name of syslog server, Transport Protocol, and Message format.	Host name of syslog server is an optional field, whereas transport protocol and message format are mandatory fields. A valid input for transport protocol is UDP or TCP. Message format can be CEF, RFC5424, LEEF, or PLAIN. The argument set is supposed to be within double quote and can be separated by comma or a space. E.g. a valid set of argument will be "syslog-host UDP PLAIN" or "syslog-host,UDP,PLAIN".
-t	--server2	Expects host name of syslog server, Transport Protocol, and Message format.	--server1 is an optional field. It is an optional field and the argument is same as the --server1 option.

-r	-server3	Expects host name of syslog server, Transport Protocol, and Message format.	It is an optional field and the argument is same as the --server1 option.
-f	--server4	Expects host name of syslog server, Transport Protocol, and Message format.	It is an optional field and the argument is same as the --server1 option.

As example, a sample command to configure a syslog server for FS type agent will be the following:

```
$ ./vmssc -d domain1 syslog addhost -a FS -h 10.3.3.163 -o "syslog-host-a UDP
PLAIN" -t "syslog-host-b UDP PLAIN" -r "syslog-host-c TCP PLAIN" -f "syslog-
host-d UDP PLAIN"
```

Syslog Showhost

The "showhost" command lists the syslog server configured for a agent type. These are the following options available with this command:

Argument	Long Form	Short Description	Notes
-h	--host	Expects a host name.	It is a mandatory field.
-a	--agenttype	Expects an argument. FS is a valid agent types.	It is a mandatory field.

A sample command to show the syslog server associated with particular configurations will be:

```
$ ./vmssc syslog showhost -a FS -h 10.3.3.163
Host Name  : 10.3.3.163
Agent Type : FS
Syslog Server settings:
syslog-host-a:
Transport Protocol : UDP
Message Format     : PLAIN
syslog-host-b:
Transport Protocol : TCP
Message Format     : PLAIN
syslog-host-c:
Transport Protocol : TCP
Message Format     : PLAIN
syslog-host-d:
Transport Protocol : TCP
Message Format     : PLAIN
```

The output of this command consists of syslog server with corresponding communication protocol and message format.

Syslog Add

The "syslog add" command adds a syslog server to DSM and specifies its setting. This syslog server is responsible for logging the system level or domain level events. These are the following options associated with this command:

Argument	Long Form	Short Description	Notes
-s	--slogserver	It requires host name or FQDN of syslog server.	It is a mandatory field.
-p	--port	Expects port number.	Port number is a mandatory field, it can be in between 1 and 65535.
-r	--protocol	Expects transport protocol as an argument.	Transport protocol is a mandatory field, it can be TCP, UDP or TLS.
-f	--format	Expects message format as an argument.	Message format is mandatory field, a valid message format can be CEF, RFC5424, LEEF or PLAIN.
-t	--timeout	Expects a value for connection time out duration.	Time-out duration is a mandatory field and it expects a valid integer value.
-c	--certificate	Expects certificate for TLS.	It is an optional field.

A sample command to add a syslog server will be the following:

```
$ ./vmssc syslog add -s test-syslog -p 8080 -r TCP -f PLAIN -t 10
```

Syslog Delete

The "syslog del" or "syslog delete" command deletes a syslog server associated with DSM (i.e.) the settings responsible for logging domain level or system level events and has been specified using "syslog add" command. Following is the syntax of this command:

```
$ ./vmssc syslog delete <host name>
```

Host name of syslog server is a compulsory argument to execute this command.

Syslog Show

The "syslog show" command shows syslog server setting from DSM. These syslog servers are logging the events at the system or domain level. A sample command to show the syslog servers associated with particular configurations will be:

```
$ ./vmssc syslog show
Syslog Server 1 Settings:
Server Name        : test-syslog
Transport Protocol : TCP
Port               : 8080
Message Format     : PLAIN
Syslog Server 2 Settings:
```

```
Server Name         : test-syslog-a
Transport Protocol : TCP
Port                : 8080
Message Format      : RFC5424
```

Backup Commands

The backup commands include the following:

Name	Description
add	Configures automatic backup.
delete	Deletes a backup schedule.
show	Shows the backup schedules.
now	Starts auto-back right away.
credentials	Shows security server credentials.

The backup commands allow an user to add, show, delete and start the back up schedules. Backup schedules can be initialized at different interval. The following sub-sections describe these commands, their usage, and examples in detail.

Backup Add

The "add" command configures and adds a backup schedule. It has the following options:

Argument	Long Form	Short Description	Notes
-t	--time	Expects time in HHMM format.	Time is a mandatory field. 24-hour time format is expected.
-D	--day	Expects day of the week.	This switch is to be used for scheduling weekly backup. Argument can be any value from sun, mon, tue, wed, thu, fri or sat.
-f	--fss	Expects file server settings.	File server setting is mandatory field and it can be scp or win. SCP requires host name, directory and user name. Similarly, win requires host name, directory name and user name for windows.
-h	--host	Requires target host name as argument.	Target host name to be used with SCP/windows share.
-d	--dir	Expects target directory.	It is to be used with SCP/Windows share.
-u	--user	Expects user name for SCP.	User name of machine.
-p	--password	Expects password.	Required for windows share.

E.g., a sample command to add a weekly backup schedule will be the following:

```
$ ./vmssc backup add -t 0030 -D sun -f scp -h test-host -d / -u root
```

Similarly, following will be the command for scheduling daily backup on 00:30:

```
$ ./vmssc backup add -t 0030 -f scp -h test-host -d ~/ -u root
```

Backup Delete

The "delete" command deletes a scheduled backup. Following is the syntax of this command:

```
$ ./vmssc backup delete
```

Backup Show

The "show" command shows the configuration of a scheduled backup. Following is the example of this command:

```
$ ./vmssc backup show
Day                   : sun
Time                  : 00:30
File Server Settings  : scp
Host Name             : test-host
Directory             : /
User Name             : root
Finger-print          : 0a:dd:ca:bd:71:cc:98:cc:f7:c5:74:0d:9a:16:07:b6
Password              : XXX
```

Based on the settings of the schedule, this command shows timing information, host related information and credentials.

Backup Now

The "backup now" command starts the backup right away. The options for this command are same as "backup add" command described above. E.g. a sample command to start backup will be the following:

```
$ ./vmssc backup now -t 0030 -D sun -f scp -h test-host -d / -u root
```

Similarly, a sample command to backup (daily) with windows share will be the following:

```
$ vmssc backup now -t 0140 -f win -h windows-host-name -d /folder_name -u
windows_user_name -p users_password
```

Backup Credentials

The "backup credentials" command shows the openssh key of DSM to be used on target machine for auto backup. Following is the syntax of this command:

```
$ ./vmssc backup credentials
Backup Credentials:
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDhu+YYWL549bzEsVWcIk2D5MLyfrJQHzq9ixAPxs+SrVZQj
mplwMNJfRenjRD6LGnRYGqqMawahOPPnwy79YqXdg42TpG635vmekBVq3BIAgfldwKmBEXUTsCfvF
8NcIPPSheuuIsHH3qd7X7UJmFjPWgs+3vzVLmq25VJUU/5Fj7dAJ/CQwKsav9b5JFTaTvWpk9LFYq
0RBuPmufvXELBQOzT2sD6SsNxxOsdH77YpVtUnIEEHqXjytmBtkhdNUfQmMgowmhHc8BUIEbHpePU
QdNZjjRt8wb/t7n/xYrT27+rwVr8X5ajr7sCwQJ+yqkR1n+aXizxCRXJbHILh40D root@vmssc-
dsm
```

Sharedsecret Commands

The sharedsecret commands include the following:

Name	Description
Set	Sets or generates registration secret.
Expire	Deletes the registration secret.
Show	Shows the registration secret.
Setlic	Sets the default license type.
Showlic	Shows the default license type.

The shared secret commands are also known as 'secret' or 'ss' commands. The sharedsecret commands allow an user to perform various operations on 'registration secret' and 'license type' for different agent types in a domain. These commands require a domain name compulsorily. The following sub-sections describe these commands in detail.

Sharedsecret Set

The "set" command sets or generates the registration secret of a domain or host group. It has the following options:

Argument	Long Form	Short Description	Notes
-v	--validity	Validity is a mandatory field.	Validity must be "hour", "day", "week", or "month. It allows the secret string to be valid for a hour, a day, a week or a month respectively.
-p	--prehost	Flag.	Flag is to be specified if it require that hosts are first added.
-h	--hg	Optional.	Expects host-group name.
-s	--secret	Optional.	Expects registration secret as argument (If it is to be added manually).

E.g., Registration secret can be set as following:

```
$ ./vmssc secret set -v day -s Ssl12345#
```

The registration secret can be generated as following:

```
$ ./vmssc secret set -v day -h test-host
$ ./vmssc secret show
Registration Secret             : vJyU!%3F
Expiration Date                 : 2015-11-10 08:30:01.0
Require hosts to be first added : No
```

Sharedsecret Expire

The "expire" command deletes the registration secret of a domain/host-group or forces it to expire. Following is the syntax of the command:

```
$ ./vmssc secret expire [host-group-name]
```

E.g.

```
$ ./vmssc secret expire host-group-1
```

Sharedsecret Show

The "show" command shows the registration secret of a given domain or host-group. Following is syntax of command:

```
$ ./vmssc secret show [host-group-name]
```

E.g.

```
$ ./vmssc secret show host-group-1
Registration Secret             : Ssl12345#
Expiration Date                 : 2015-11-10 08:30:01
Require hosts to be first added : Yes
```

Sharedsecret Setlic

The "setlic" command sets the default license type of a agent. These are the following options in this command:

Argument	Long Form	Short Description	Notes
-a	--agent	Agent name is a mandatory field.	Valid agent types are FS, DB2, IDS, KEY or KMIPC.
-l	--license	License type is mandatory field.	Valid license types are TERM, PERPETUAL or HOURLY.

E.g.

```
$ ./vmssc secret setlic -a FS -l TERM
```

Sharedsecret Showlic

The "showlic" command shows the default license type of a domain. Following is the syntax of command:

```
$ ./vmssc secret showlic <agent type>
```

Valid agent types for this command are FS, DB2, IDS, KEY or KMIPC.

E.g.

```
$ ./vmssc secret showlic FS
License Type : Term
```