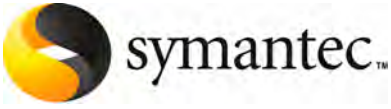
Veritas™ Cluster Server One Command Reference Guide



Veritas™ Cluster Server One Command Reference Guide

The software described in this book is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

Product version: 5.0 Service Pack 1

Documentation version: 5.0.SP1.0

Legal Notice

Symantec Corporation
350 Ellis Street
Mountain View, CA 94043

http://www.symantec.com

Technical Support

Symantec Technical Support maintains support centers globally. Technical Support's primary role is to respond to specific queries about product features and functionality. The Technical Support group also creates content for our online Knowledge Base. The Technical Support group works collaboratively with the other functional areas within Symantec to answer your questions in a timely fashion. For example, the Technical Support group works with Product Engineering and Symantec Security Response to provide alerting services and virus definition updates.

Symantec's support offerings include the following:

- A range of support options that give you the flexibility to select the right amount of service for any size organization
- Telephone and/or web-based support that provides rapid response and up-to-the-minute information
- Upgrade assurance that delivers software upgrades
- Global support purchased on a regional business hours or 24 hours a day, 7 days a week basis
- Premium service offerings that include Account Management Services

For information about Symantec's support offerings, you can visit our web site at the following URL:

www.symantec.com/business/support/

All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policy.

Contacting Technical Support

Customers with a current support agreement may access Technical Support information at the following URL:

www.symantec.com/business/support/

Before contacting Technical Support, make sure you have satisfied the system requirements that are listed in your product documentation. Also, you should be at the computer on which the problem occurred, in case it is necessary to replicate the problem.

When you contact Technical Support, please have the following information available:

- Product release level

- Hardware information
- Available memory, disk space, and NIC information
- Operating system
- Version and patch level
- Network topology
- Router, gateway, and IP address information
- Problem description:
  - Error messages and log files
  - Troubleshooting that was performed before contacting Symantec
  - Recent software configuration changes and network changes

Licensing and registration

If your Symantec product requires registration or a license key, access our technical support web page at the following URL:

www.symantec.com/business/support/

Customer service

Customer service information is available at the following URL:

www.symantec.com/business/support/

Customer Service is available to assist with non-technical questions, such as the following types of issues:

- Questions regarding product licensing or serialization
- Product registration updates, such as address or name changes
- General product information (features, language availability, local dealers)
- Latest information about product updates and upgrades
- Information about upgrade assurance and support contracts
- Information about the Symantec Buying Programs
- Advice about Symantec's technical support options
- Nontechnical presales questions
- Issues that are related to CD-ROMs or manuals

Support agreement resources

If you want to contact Symantec regarding an existing support agreement, please contact the support agreement administration team for your region as follows:

Asia-Pacific and Japan	customercare_apac@symantec.com
Europe, Middle-East, and Africa	semea@symantec.com
North America and Latin America	supportsolutions@symantec.com

Additional enterprise services

Symantec offers a comprehensive set of services that allow you to maximize your investment in Symantec products and to develop your knowledge, expertise, and global insight, which enable you to manage your business risks proactively.

Enterprise services that are available include the following:

Managed Services	Managed Services remove the burden of managing and monitoring security devices and events, ensuring rapid response to real threats.
Consulting Services	Symantec Consulting Services provide on-site technical expertise from Symantec and its trusted partners. Symantec Consulting Services offer a variety of prepackaged and customizable options that include assessment, design, implementation, monitoring, and management capabilities. Each is focused on establishing and maintaining the integrity and availability of your IT resources.
Education Services	Education Services provide a full array of technical training, security education, security certification, and awareness communication programs.

To access more information about enterprise services, please visit our web site at the following URL:

www.symantec.com/business/services/

Select your country or language from the site index.

Contents

Chapter 1

Veritas Cluster Server One commands overview

This chapter includes the following topics:

Commands overview

Veritas Cluster Server One (VCS One) commands enable administrators and operators to manage the VCS One cluster from the command line. The product requires that you have sufficient privileges to manage the VCS One cluster configuration and the VCS One objects. The VCS One objects include systems, resources, service groups, and users.

About VCS One commands

Managed applications running in a VCS One cluster are resources consolidated into service groups. The VCS One cluster, systems, service groups, and resources that compose them, are all considered VCS One objects. In VCS One, the users who administer and manage the objects are themselves objects.

The commands in Table 1-1 enable the management of VCS One objects.

Table 1-1 VCS One Commands

Command	Tasks
haadmin	Administer the Policy Master service group (PMSG) in the Policy Master cluster. See haadmin on page 19.
haagent	Administer the agents that control resources. See haagent on page 27.
haat	Administer authentication. See haat on page 31.
haattr	Define, add, and remove attributes and default values; display VCS One object attribute values. See haattr on page 64.
haclus	Manages the VCS One cluster. See haclus on page 73.
haconf	Manage the VCS One configuration, including loading the configuration from files or a database and converting the configuration from one form to another. See haconf on page 78.
hacsg	Manages composite service groups. See hacsg on page 83.
hadb	Manages the VCS One configuration database. See hadb on page 94.
haea	Create and maintain extended attributes. See haea on page 99.
haencrypt	Generate encrypted passwords for VCS One configurations. See haencrypt on page 106.
hapframe	Add, modify, or delete the physical systems that you use exclusively for virtualization (these systems are called "frames"). Display or list information about frames. See hapframe on page 108.

Table 1-1 VCS One Commands *(continued)*

Command	Tasks
havframe	Add, modify, or delete a virtual machine; display or list information about virtual machines. See havframe on page 117.
havobject	Add, modify, delete, display, and list vobjects; display the attribute value for a given vobject. See havobject on page 143.
hagetcf	Create a gzip file that contains log files, and information about your configuration and systems. You can then send this gzip file to Symantec Technical Support so that they can troubleshoot issues with your VCS One configuration. See hagetcf on page 149.
hagrp	Manage service groups and define how they work within the VCS One cluster and with other service groups. See hagrp on page 152.
hagtq	Manage the VCS One Group Transition Queue (GTQ). See hagtq on page 176.
haldapconf	Configure LDAP. See haldapconf on page 180.
halog	Add messages to the engine log. See halog on page 185.
halogin	Provide credentials to authenticate VCS One users. See halogin on page 188.
hamultisim	Create and use multiple Simulator instances. See hamultisim on page 192.
haou	Create and maintain the Organization Tree. See haou on page 196.
hares	Manage service group resources. See hares on page 201.

Table 1-1 VCS One Commands *(continued)*

Command	Tasks
harole	Create roles based on a combination of VCS One objects with operation privilege levels. See harole on page 212.
harule	Create and manage rules. See harule on page 220.
haset	Create and maintain sets. See haset on page 226.
hasim	Start and stop the VCS One Simulator. Using the command line, simulate faults of systems, pframes, vframes, resources, and service groups in a VCS One cluster. See hasim on page 230.
hastart	Start the Policy Master service group and, if disaster recovery is configured, the disaster recovery service group. The `-hastart` command also starts the Policy Master cluster, VCS One configuration database, the Policy Master daemon and VCS One client daemons. See hastart on page 236.
hastatus	View the status of the VCS One cluster and VCS One objects. See hastatus on page 239.
hastop	Stop the VCS One Policy Master and VCS One client daemons, stop the Policy Master service group, stop the VCS One database, or stop the web console. See hastop on page 242.
hasys	Manage the VCS One cluster systems. See hasys on page 248.
hatype	Manage the VCS One resource types that control specific resources. See hatype on page 257.
hauser	Add and remove VCS One users and manage their privileges. See hauser on page 262.

Table 1-1 VCS One Commands *(continued)*

Command	Tasks
havtype	Manage the VCS One vtypes that control specific frames. See havtype on page 268.
vxfentsthdw	Test storage devices for SCSI-3 reservations compliance. See vxfentsthdw on page 272.

Setting the PATH variable to use the command line interface on UNIX

Both VCS and VCS One are installed on the Policy Master. Sometimes, the same command is in both projects; for example, `halog` and `haclus`.

To avoid confusion, when you execute a command, use the full path name.

To set the PATH variable to use the command line interface (CLI) with VCS One

1. If you have previously set the path variable for VCS, remove /opt/VRTSvcs/bin from it.
2. At the command prompt, enter the following:

```
PATH=$PATH:/opt/VRTSvcsone/bin export PATH
```

Specifying the command path on Windows

Both VCS and VCS One are installed on the Policy Master. Sometimes, the same command is in both products; for example, `halog` and `haclus`.

On Windows, the installation process sets a path variable for both VCS and VCS One.

If you execute a command that exists in both products, the operating system runs the VCS command if it locates that path first.

To avoid executing a VCS One command in VCS, use the CD command to change the directory to the following path:

```
%VCSONE_HOME%\bin
```

where

`%VCSONE_HOME%` is the path that you specify during VCS One installation. The default installation path is:

```
C:\Program Files\Veritas\Cluster Server One
```

About UNIX online manual pages

The VRTSvcsonemn package includes online manual pages. These man pages are installed in the appropriate directories under `/opt/VRTS/man`. Add this path to the MANPATH environment variable for your platform.

On Windows, the installation does not include online manual pages.

See Table 1-2 on page 14. describes how to set the MANPATH environment variable for your UNIX platform.

See "Specifying the command path on Windows" on page 13. for instructions on how to specify the command path on Windows.

Table 1-2 How to set the MANPATH environment variable

Platform	How to set the MANPATH

SUSE Linux Enterprise Server 9 (SLES 9)	Add the following lines to `/etc/man.config`:
	`MANPATH /opt/VRTS/man`
	`MANPATH_MAP /opt/VRTSvcsone/bin /opt/VRTS/man`
	Also, add "1m" to the existing SECTION line:
	`SECTION 1 n l 8 3 2 5 4 9 6 7 1x 3x 5x 6x 1m`

RedHat Enterprise Linux (RHEL)	Add the following lines to `/etc/man.config`:
	`MANPATH /opt/VRTS/man`
	`MANPATH_MAP /opt/VRTSvcsone/bin /opt/VRTS/man`
	Also, add "1m" to the existing MANSECT line:
	`MANSECT 1:8:2:3:4:5:6:7:9:tcl:n:l:p:0:1m`

Solaris, HP-UX, and AIX	Run one of the following commands:
	`export MANPATH=$MANPATH:/opt/VRTS/man`
	`setenv MANPATH {$MANPATH}:/opt/VRTS/man`
	Note: To configure this environment variable so that it applies every time you log on , add the `export` or `setenv` command to your .login or .cshrc file.

Setting the MANPATH environment variable does not update the windex database. To make sure that VCS One manual pages display correctly after you install VCS One, update the windex database.

Appendix A

Veritas Cluster Server One commands

This appendix includes the following topics:

- haldapconf
- halog
- halogin
- hamultisim
- haou
- hares
- harole
- harule
- haset
- hasim
- hastart
- hastatus
- hastop
- hasys
- hatype
- hauser
- havtype
- vxfentsthdw

haadmin

`haadmin` – enables switch, freeze, unfreeze, clear, and other operations for the Policy Master service group (PMSG) and the disaster recovery service group (DRSG)

SYNOPSIS

```
UNIX: /opt/VRTSvcsone/bin/haadmin

Windows: %VCSONE_HOME%\bin\haadmin

haadmin -status [-summary]
haadmin -state [-sys system]
haadmin -switch -to system
haadmin -freeze [-persistent]
haadmin -unfreeze [-persistent]
haadmin -clear [-sys system]
haadmin -backup [-vss | -db [-incremental]] backup_dir
haadmin -restore [-vss | -db] backup_dir
On Linux: haadmin -addnic -niclistfile niclistfilename -netmask
netmask [-ipmp]
On Solaris: haadmin -addnic nic1 [nic2 nic3...] [-ipmp]
On Windows: haadmin -addnic System MACAddress
On Linux and Solaris: haadmin -displaynic
haadmin -deletenic nic
haadmin -displaynic
haadmin adddrnic nic
haadmin -deletedrnic nic
haadmin -addip ip_address nic netmask [-port port]
haadmin -adddrip ipaddress nic netmask
haadmin -deleteip ip_address [-port port]
haadmin -deletedrip ipaddress
haadmin -displayip
haadmin -version
haadmin [-help]
For the Simulator, the command usage is:
haadmin -backup -db backup_dir
haadmin -restore -db backup_dir
haadmin [-help]
```

AVAILABILITY

VRTSvcsonepm

DESCRIPTION

`haadmin` administers the Policy Master service group (PMSG) and the disaster recovery service group on the VCS One Policy Master cluster. The PMSG and DRSG are not VCS One service groups. The Policy Master cluster uses Veritas Cluster Server (VCS) to provide high-availability for the Policy Master service group and the disaster recovery service group. VCS controls and monitors the resources of the PMSG and DRSG.

The tasks you can perform with `haadmin` include the following:

- View the service group state
- View the status of service group resources
- Freeze and unfreeze resources
- Switch the service group to another system in the Policy Master cluster
- Clear service group faults
- Add secondary IP addresses to the PMSG configuration
- Remove secondary IP addresses from the PMSG configuration

See "OPTIONS" on page 20.

`-backup` Backs up all authentication service configuration data and all Policy Master database information to a safe location. `-restore` Restores the configuration data and database data you backed up using the `-backup` option. With these options, you can prepare for recovery from the possible loss of both Policy Master systems and the authentication service and database configuration data.

OPTIONS

`-status [-summary]`

`-status` Displays the status of resources of the Policy Master service group (and the DRSG, if disaster recovery is configured) on each system. `-summary` Displays a condensed version of the status.

`-state [-sys` *`system`*`]`

`-state` Displays the state of the Policy Master service group (and the DRSG, if disaster recovery is configured). `-sys` *system* Displays the state of a specific system.

`-switch -to` *system*

Switches the Policy Master service group (and the DRSG, if disaster recovery is configured) to the specified system.

`-freeze [-persistent]`

`-freeze` Disables any online offline, or failover operations on the Policy Master service group. In a disaster recovery configuration, `-freeze` also disables operations on the DRSG. `-persistent` Specifies that the frozen state continues after you restart the Policy Master cluster.

`-unfreeze [-persistent]`

`-unfreeze` Resumes any online, offline, or failover operations on the Policy Master service group. In a disaster recovery configuration, `-unfreeze` also unfreezes the DRSG. `-persistent` Continues the unfrozen state after you restart the Policy Master cluster.

`-clear [-sys` *system*`]`

`-clear` Changes a Policy Master service group's fault (and a disaster recovery service group's fault, if disaster recovery is configured) by changing the resource states from faulted to offline. If you do not specify a system, the option affects all resources on all systems in the group's system list. `-sys` Clears the fault for the Policy Master service group (or the disaster recovery service group, if configured) on the specified system.

`-backup [-vss | -db [-incremental]]` *backup_dir*

`-backup` Copies all security and Policy Master database data and configuration information to a specified directory. The Policy Master database must be up and running when you back it up. When you back up the security-related information, make sure the VxAT process is running. Security-related information is backed up to the vcsone_vxssbackup.tar file. If a file named vcsone_vxssbackup.tar is in the directory, it is renamed with the suffix .old.

Use the `-vss` command as follows:

- On UNIX, `-vss` only backs up security-related information. The authentication service configuration data is on shared storage. Back it up and restore it from the active Policy Master system.
- On Windows `-vss` backs up the entire directory to the backup directory location you specify. Make sure that you back up to a separate drive. Avoid backing up to the system (C:) drive, since a system crash can make the backup data unavailable.

`-db` Backs up the Policy Master database to a specified directory. Unless you specify `-incremental`, `-db` backs up the entire database. `-incremental` Only copies the parts of the database configuration that have changed since the

last backup. You can only use this option with `-db`; you cannot use it to back up security-related data.

For more information on backup and restore operations, see the *Veritas Cluster Server One User's Guide*.

`-restore [-vss | -db] backup_dir`

`-restore` Restores all security data, database data, and configuration information from the specified directory.

After you use the `-restore` command option, VCS One and CLI commands no longer work. To resolve this issue, run the following command on every Policy Master node:

```
/opt/VRTSvcsone/bin/haat setuptrust -b PM_VIP:BrokerPort -s low
```

`-vss` Restores the security-related information about the Policy Master. The restored information comes from the backup tar file. Make sure that the VxAT process is running and that you have mounted the shared storage where the security-related information is to be restored. After you restore the security information, restart the VxAT process.

The authentication service configuration data is stored on shared storage. Therefore, it needs to be backed up and restored from the active Policy Master system.

`-db` Restores the Policy Master cluster database from the specified backup directory.

`-addnic -niclistfile niclistfilename -netmask netmask [-ipmp]`

`-addnic` On Linux, adds the specified NIC or NICs.

The *niclistfilename* must contain a list of NICs and their base addresses in the following format:

#SystemList	name_of_sys1	name_of_sys2...	name_of_sysN
nic1	baseip1_on_sys1	baseip1_on_sys2	baseip1_on_sysN
nic2	baseip2_on_sys1	baseip2_on_sys2	baseip2_on_sysN
nic3	baseip3_on_sys1	baseip3_on_sys2	baseip3_on_sysN

If nic1 is configured under an existing MultiNICA resource, VCS One adds NICs (nic2, nic3, and so on) to that resource.

If nic1 is not part of a NIC or MultiNICA resource, VCS One creates a new secondary MultiNICA resource and calls it `pmsecnicn`.

On Linux, `-ipmp` is ignored.

`-addnic` *`nic1`* `[`*`nic2 nic3 …`*`] [-ipmp]`

`-addnic` On Solaris, adds the specified NIC, nic1.

If nic1 is configured under an existing MultiNICB resource, VCS One adds NICs (nic2, nic3, and so on) to the resource.

If nic1 is not part of a NIC or MultiNICB resource, VCS One creates a new secondary MultiNICB resource called pmsecnicn.

`-ipmp` Uses the Solaris IP multipathing mode with the MultiNICB agent. If you do not specify `-ipmp`, the VCS MultiNICB mode is used.

`-addnic` *`System MACAddress`*

On Windows, `-addnic` adds the specified NIC. VCS One Windows does not support a multiple NIC resource. Each IP address depends on one NIC. For every NIC, a NIC resource is added to the Policy Master service group. NIC resources are named `pmnic$index`, where `pmnic1` is the primary NIC that you specify using the VCS One Policy Master Configuration Wizard. The primary NIC cannot be deleted from the configuration. Any additional NICs you add will be named pmnic2, pmnic3, and so on. You can add only TCP/IP enabled MACs.

`-deletenic` *`nic`*

Deletes the specified NIC.

`-displaynic`

Displays all NICs. This option is the same on Linux and Solaris.

`-adddrnic` *`nic`*

Adds a resource that is associated with the specified NIC device to the disaster recovery service group (DRSG). The DRSG is configured as part of the disaster recovery (DR) configuration using the installer.

On Windows, the `-adddrnic` option is not supported.

`-deletedrnic` *`nic`*

Deletes the resource that is associated with the specified NIC device from the disaster recovery service group (DRSG).

On Windows, the `-deletedrnic` option is not supported.

`-addip` *`ip_address nic netmask`* `[-port` *`port`*`]`

Adds an IP address to the PMSG and update the VCS One Policy Master with the new IP address on which to listen.

On Windows, there is no multiple NIC resource, so each IP depends on one NIC. The *nic* value is the name (not the MAC address) of a NIC added using the `-addnic` option. To get the NIC name, you can use the `-displaynic` option.

`-adddrip` *`ip_address nic netmask`*

Adds the IP address resource in the DRSG. The Policy Master starts listening on this IP address for a disaster recovery connection. *nic* is the NIC device used for the disaster recovery IP address.

On Windows, the `-adddrip` option is not supported.

vcsoned must be running on the system where you use this command.

`-deleteip` *`ip_address`* `[-port` *`port`*`]`

Deletes an IP address from the PMSG. You cannot delete the primary IP address, the IP address on which other resources depend.

`-deletedrip` *`ip_address`*

Deletes the IP address resource from the DRSG, but the Policy Master does not stop listening on the IP address. After a failover, the Policy Master stops listening on the IP address.

vcsoned must be running on the system where you use this command.

If the IP address is online, run `ifconfig down` manually for the IP address after deleting the IP address resource using `haadmin -deletedrip`. `haadmin -deletedrip` does not run `ifconfig down` for an IP address that is online.

On Windows, the `-deletedrip` option is not supported.

`-displayip`

Lists the IP resources for the PMSG (and the DRSG, if disaster recovery is configured) and the IP addresses for those resources.

`-version`

Displays `haadmin` command version.

`[-help]`

Describes how to use the `haadmin` command.

The following command options apply for the Simulator:

`-backup -db` *`backup_dir`*

In the Simulator, this command option backs up all security and Policy Master database data and configuration information to a specified back-up directory. To back up the Policy Master, it must be up and running when you issue this command.

`-restore -db backup_dir`

In the Simulator, this command option restores all security data, database data, and configuration information from a specified back-up directory.

`[-help]`

Displays usage for the `haadmin` command.

EXAMPLES

To check the status of the PMSG on each system:

```
haadmin -status
```

To get a summarized version of the status of the PMSG:

```
haadmin -status -summary
```

To switch the PMSG to another system (system1) in the Policy Master cluster:

```
haadmin -switch -to system1
```

To incrementally back up the database on Linux or Solaris, enter the following command:

```
haadmin -backup -db -incremental /var/tmp
```

To incrementally back up the database on Windows, enter the following command:

```
haadmin -backup -db -incremental D:\Temp
```

To add a NIC on Linux:

```
haadmin -addnic niclistfile /root/addniclist.txt -netmask 255.255.255.0
```

The addniclist.txt file contains the following information:

```
#SystemList sys1

eth0 192.168.100.200

eth1 192.168.100.201
```

To add a NIC on Solaris without the IPMP feature:

```
haadmin -addnic bge0 bge1
```

To add a NIC on Solaris with the IPMP feature:

```
haadmin -addnic bge0 bge1 -ipmp
```

To add a NIC on Windows:

```
haadmin -addnic Sys1 00-50-56-14-00-01 Sys2 00-0C-29-8D-9C-E4
```

To add a disaster recovery NIC:

```
haadmin -adddrnic eth1
```

To delete a disaster recovery NIC:

```
haadmin -deletedrnic eth2
```

To add an IP address with a customized port:

```
haadmin -addip 192.168.100.200 bge0 255.255.255.0 -port 12321
```

To add a disaster recovery IP address:

```
haadmin -adddrip 10.182.11.154 eth2 255.255.244.0
```

To delete a disaster recovery IP address:

```
haadmin -deletedrip 10.182.1.153
```

To display a list of the IP resources and addresses in the PMSG, enter the following command.

```
 haadmin -displayip
```

SEE ALSO

`hastart`(1M), `hastop`(1M), `hadb`(1M)

haagent

`haagent` – administer the agents and the processes that manage VCS One resources

SYNOPSIS

```
UNIX: opt/VRTSvcsone/bin/haagent

Windows: %VCSONE_HOME%\bin\haagent

haagent -start agent -sys system [-user user@domain -domaintype
domaintype]
haagent -stop [-notransition] agent -sys system [-user user@domain
-domaintype domaintype]
haagent -dumpffdc agent -sys system [-user user@domain -domaintype
domaintype]
haagent -display [agent(s)] [-attribute attribute(s)] [-sys system(s)]
[-user user@domain -domaintype domaintype]
haagent -list [conditional(s)] [-sys system(s)] [-user user@domain
-domaintype domaintype]
haagent -value agent attribute [-sys system(s)] [-user user@domain
-domaintype domaintype]
haagent -update agent [-user user@domain -domaintype domaintype]
haagent -update -all [-user user@domain -domaintype domaintype]
haagent [-help [-list]]
haagent -version
```

AVAILABILITY

VRTSvcsonec

DESCRIPTION

The `haagent` command starts, stops, displays, and lists VCS One agents. You may also use the command to dump FFDC logs for a specified agent.

The `-start` and `-stop` options enable you to debug custom agents without having to start and stop the VCS One client daemon.

A non-root user who has not run the `halogin` command can execute the `haagent` command using the `-user` *user@domain* option. This option executes the command with the privileges of the specified user. When you issue the command, enter your

fully qualified domain user name and supply a password when prompted. If necessary, the `-domaintype` option can specify the type of domain against which the user is to be authenticated. Supported domain types include:

- "unixpwd"
- "nt"
- "nis"
- "nisplus"
- "ldap"
- "pam"
- "vx" (Symantec Private Domain)

The default domain type is "vx". The domain type is case sensitive.

When using `domaintype=unixpwd`, provide the system name as the domain portion. The domain must be a fully-qualified domain name (for example, sun01.engba.veritas.com).

See NOTES for how to specify "-" and "%" characters in the command line.

OPTIONS

`-start` *`agent`* `-sys` *`system`*

Manually start the specified agent on the specified system. This command is required only if the agent is stopped. Otherwise, the VCS One client daemon automatically starts an agent if a resource for the corresponding agent is configured for the specified system.

`-stop [-notransition]` *`agent`* `-sys` *`system`*

Manually stop the specified agent on the specified system. Use the `-notransition` option to manually stop the agent when all resources are in a stable state. Resources are in a stable state when there are no resources that are in any of the following states:

- Offline and waiting to go online
- Online and waiting to go offline
- Restarting on the specified system

If there are no resources that are in any of the previous states, the agent stops, and all resources are left in their current state. For example, the resources that are online are left online, and the resources that are offline are left offline.

`-dumpffdc` *agent* `-sys` *system*

Dumps first-failure data capture (FFDC) logs for the specified agent to /var/VRTSvcsone/diag/agents/*agent*. The format of FFDC log files is FFDC_*role_PID_agent*.log, where *role* is AGFWMain, AFGWSvc, or AGFWTimer, *PID* is the process identification number, and *agent* is the agent name.

For example, if the PID of the FileOnOff agent is 18602, the command:

```
haagent -dumpffdc FileOnOff -sys system
```

The output resembles the following:

```
# ls -1 /var/VRTSvcsone/diag/agents/FileOnOff
FFDC_AGFWMain_18602_FileOnOff.log FFDC_AGFWSvc_18602_FileOnOff.log
FFDC_AGFWTimer_18602_FileOnOff.log
```

You may change the dump file location by setting the VCSONE_DIAG environment variable to the desired location. You may disable the dumping of FFDC logs by setting the VCSONE_DISABLE_FFDC_ON_BOOT environment variable. You may enable FFDC log dumping by unsetting it. Non-root users with the role type S_DumpFFDCAgent can use the `-dumpffdc` command option.

`-display [`*agent(s)*`] [-attribute` *attribute(s)*`] [-sys` *system(s)*`]`

Display information about all agents or about a specified agent. Use the `-attribute` option to specify the display of a resource attribute. The command displays agent information for the local system if a system is not specified.

`-list [`*conditional(s)*`] [-sys` *system*`]`

Displays a list of agents whose values match given conditional statement(s). Conditional statements can take three forms: Attribute=Value, Attribute!=Value, Attribute=~Value. Multiple conditional statements imply AND logic. All agents that are configured on the local system are listed by default. If a system is specified, the agents that are configured on the specified system are displayed. Conditionals can be used to list only those agents that meet the conditional criteria.

`-value` *agent attribute* `[-sys` *system(s)*`]`

The `-value` option provides the value of a single agent attribute. For example, `haagent -value Mount Running` displays the value of the Running attribute for the Mount agent. The -value option is used instead of the `-display` option when one specific attribute value is needed rather than a table of many attribute values. The command displays agent information for the local system if a system is not specified.

`-update` *`agent`*

`-update` *`agent`* Parses the *agent*.xml on the local system and send the agent version to the Policy Master.

`-update -all`

`-update -all` Parses the *agent*.xml files for all agents that are defined for the current system, and sends the agent version information to the Policy Master. If the agent version information cannot be determined, the version is reported as UNKNOWN.

`[-help [-list]]`

Displays usage information about the `haagent` command. The `-list` option provides the usage for the `list` option. When you enter the command and an option without arguments, syntax for the specific option displays.

`-version`

Displays command version information.

EXAMPLES

To display the usage syntax for a specific command option, enter the command and an option without arguments. For example, to display usage information for `haagent -value`, enter:

```
# haagent -value
```

NOTES

When using the command to specify or modify an attribute value that begins with a dash ("-"), precede the value with a percent sign ("%"). For example, specify `-y` as `%-y`. Likewise, precede a value that starts with a percent sign with another percent sign.

SEE ALSO

`halogin`(1M)

haat

`haat` – manages Symantec Product Authentication Service (AT)

SYNOPSIS

UNIX: opt/VRTSvcsone/bin/haat

Windows: %VCSONE_HOME%\bin\haat

haat *options*
The options for the haat command are listed below. Each option has suboptions. The options and suboptions are explained in the OPTIONS section.
Commonly-used client-side options are:
authenticate
importrootcred
setuptrust
showcred
showversion
Commonly-used broker-side options are:
addprpl
authenticate
createpd
importrootcred
setuptrust
showcred
showversion
Options for remote identity deployment are:
showcredinfo
Options for broker administration are:
addauthsequence
addldapdomain
createpd
deleteauthsequence
deletepd
listldapdomains
listpd
removeldapdomain
setexpiryintervals
setispbxexchflag

```
setpd
setpdr
showauthsequence
showbackuplist
showbrokerhash
showbrokermode
showbrokertag
showdomains
showexpiryintervals
showglobalplugininfo
showispbxexchflag
showpd
showpdr
showplugininfo
showrootbroker
updateplugin
Options for remote administration are:
addprpl
changepasswd
createpd
deletecred
deleteprpl
listpdprincipals
renewcredential
resetpasswd
showprpl
updateprpl
validategroup
validateprpl
Options for principal administration are:
addprpl
changepasswd
deletecred
deleteprpl
listpdprincipals
renewcredential
resetpasswd
showprpl
updateprpl
validategroup
```

```
validateprpl
Other options are:
checkclockskew
deletebrokerdomain
deleteexpiredcreds
deleteexpiredsessions
exportrootcred
getbrokeruuid
login
logout
pullbrokerattribs
pushbrokerattribs
refreshtrust
removesessioncache
removetrust
restorebroker
setbrokerlog
setclockskewtolerance
setcredstore
setdomaindiscoveryinterval
setloglevel
setmaxlogfiles
setmaxlogfilesize
setsecuritylevel
setsessioncacheparams
setsystemtrustdir
settrustrefreshparams
showallbrokerdomains
showbrokers
showclockskewtolerance
showcredstore
showalltrustedcreds
showdomaindiscoveryinterval
showsecuritylevel
showsessioncacheparams
showsystemtrustdir
showtrustrefreshparams
whoami
To view command usage for any option, enter:
haat option_name -help
```

```
To view a list of all client-side command options, enter:
haat all -help
To view a list of all broker-side command options, enter:
haat all -help -j broker
To view a list of command options for remote administration, enter:
haat remoteadmin -help
```

AVAILABILITY

VRTSvcsonec

DESCRIPTION

Use the `haat` command to administer Symantec Product Authentication Service in Veritas Cluster Server One.

OPTIONS

The command options for `haat` are listed alphabetically.

`addauthsequence -a` *`plugin_name`*

Adds one or more plug-ins at the end of the authentication sequence. You can also use this command to set an entirely new authentication sequence or append new plug-ins at the end of the sequence.

The default authentication sequence is "pam unixpwd nisplus nis."

`-a, --add` *`Plugin Name`*

Specifies the name of the plug-in to be added.

`addldapdomain -d` *`domain_name`* `-s` *`server_URL`* `-u` *`user_base_DN`* `-g` *`group_base_DN`* `[-f` *`trusted_CA_file_name`*`] [-t rfc2307|msad] | [-c` *`user_object_class`* `-a` *`user_attribute`* `-q` *`user_GID_attribute`* `-x` *`group_object_class`* `-y` *`group_attribute`* `-z` *`group_GID_attribute`*`] [-k DN|UID] [-b FLAT|BOB|FLAT SKIPNESTED|BOB SKIPNESTED] [-m` *`admin_user_DN`*`] [-w` *`admin_user_password`*`] [-p SUB|ONE|BASE]`

Adds an LDAP domain to the authentication broker. If you are not familiar with how LDAP operates, work with your LDAP administrator to determine the following information:

- The type of LDAP directory the enterprise uses. For example, Active Directory or OpenLDAP.
- The URL for the LDAP directory. For example: ldap://my_ldap_host.mydomain.myenterprise.com:389

An LDAP URL starts with "ldap://" for non-SSL or "ldaps://" for SSL-enabled LDAP.

- The distinguished name (DN) of the users container. Normally, the users container is in one of the naming contexts. For most LDAP directories, you can use the `ldapsearch` utility, provided by the directory vendor, to find out the naming contexts. For example:

  ```
  ldapsearch -x -h my_host -s base -b "" namingContexts
  ```

  For Active Directory, the users container resembles:

  ```
  cn=users,dc=domain_name,dc=enterprise_name,dc=com
  ```

- The distinguished name (DN) of the groups container. Normally, the groups container is in one of the naming contexts.
- The schema to facilitate users and groups.
  If the enterprise has migrated its NIS data to the LDAP directory according to Request for Comments 2307, it must use the RFC 2307 schema. RFC 2307 uses the posixAccount objectclass to facilitate user objects. It uses the posixGroup objectclass to facilitate group objects. If the enterprise uses Active Directory, it must use the Active Directory schema. In this schema, the user objectclass facilitates both user objects and group objects. If the enterprise uses neither RFC 2307 nor Active Directory, determine the following:
- The LDAP objectclass to facilitate user objects
- The LDAP objectclass to facilitate group objects
- The user attribute in the user objectclass to facilitate user name/ID. Use the following rules to construct the DN to the user entry:
  *user_attribute=user_name,user_container_DN*In the following example, the user attribute is configured to `cn` and the user's container DN is configured to:

  ```
  dc=mydomain,dc=myenterprises,dc=com
  ```

  The user name for the authenticate call is `jdoe`, and the LDAP DN for `jdoe` is:

  ```
  cn=jdoe,dc=mydomain,dc=myenterprise,dc=com
  ```

- The group identifier (GID) attribute that identifies the groups the given user belongs to. The GID is in the user objectclass.

- The group attribute in the group objectclass to facilitate group name. The following rules are used to construct the DN to the group entry: `group_attribute=group_name, group_container_DN`. In the following example, the group attribute is configured to `cn` and the group's container DN is configured to:

  ```
  dc=mydomain,dc=myenterprise,dc=com
  ```

  The group name is `adm`, the LDAP DN for `adm` is:

  ```
  cn=adm,dc=mydomain,dc=myenterprise,dc=com
  ```

- The group ID attribute in the group objectclass to facilitate group ID for the given group.

  `-d, --domain` *DomainType*:*DomainName*

  Specifies a symbolic name that uniquely identifies an LDAP domain.

  `-s, --server_url` *Server URL*

  Specifies the URL of the LDAP directory server for the given domain. The LDAP server URL must start with either "ldap://" or "ldaps://". Starting with "ldaps://" indicates that the given LDAP server requires SSL connection. If the LDAP server URL starts with "ldaps://", specify `-f`.

  `-u, --user_base_dn` *User Base DN*

  Specifies the LDAP-distinguished name for the user container. For example,

  ```
  ou=user,dc=mydomain,dc=myenterprise,dc=com
  ```

  `-g, --group_base_dn` *Group Base DN*

  Specifies the LDAP-distinguished name for the group container. For example,

  ```
  ou=group,dc=mydomain,dc=myenterprise,dc=com
  ```

  `-f, --server_trusted_ca_file` *Trusted CA file Name*

  Specifies the complete path to the file that contains the trusted CA certificates in PEM format. Use this parameter if the given LDAP server URL starts with "ldaps://" (indicating the need for an SSL connection). If the given LDAP server URL, however, starts with "ldap://", omit this parameter.

  `-t, --schema_type` *Schema Type*

  Specifies the type of LDAP schema.

  If you use `-t`, omit the following parameters: `-c`, `-a`, `-i`, `-o`. These values are set automatically, based on the schema type. If you do not use `-t`,

neither the rfc2307 nor the msad parameters are set automatically (you provide the values).
Two default schemas are supported:

- rfc2307: The schema that is specified in RFC 2307
- msad: Microsoft Active Directory schema

For the msad schema, if you select the BOB authentication type, the user attribute is set to sAMAccountName.

`-c, --user_object_class` *`User Object Class`*

Specifies the LDAP object class for the user object (that is, posixAccount).

`-a, --user_attribute` *`User Attribute`*

Specifies the user attribute within the user object class, using the following syntax:

`user_attribute=principal_name,user_base_DN`

For example, the LDAP DN for `jdoe` is as follows:

```
cn=jdoe,dc=mydomain,dc=myenterprise,dc=com
```

Where the *user_attribute* is `cn`, the *principal_name* is `jdoe`, and the *user_base_DN* is `dc=mydomain,dc=myenterprise,dc=com`.

Do not use the `-a` option if you use `-t`.

-q, --user_gid_attribute *User Group ID Attribute*

Specifies the attribute within the user object class to retrieve the groups the user belongs to. Do not use this option if you use `-t`.

`-x, --group_object_class` *`Group Object Class`*

Specifies the LDAP object class for the group object (that is, posixGroup). Do not use this option with `-t`.

`-y, --group_attribute Group Attribute`

Specifies the group attribute within the group object class, using the following syntax: `group_attribute=group,group_base_DN`

For example, the LDAP DN for `adm` is as follows:

```
cn=adm,dc=mydomain,dc=myenterprise,dc=com
```

Where the *group_attribute* is `cn`, the *group* is `adm`, and the *group_base_DN* is `dc=mydomain,dc=myenterprise,dc=com`.

Do not use the `-y` option if you use `-t`.

`-z, --group_gid_attribute` *`Group GID Attribute`*

Specifies the attribute within the group object class to retrieve the group. Do not use the `-z` option if you use `-t`.

`-k, --group_gid_attribute_type` *`Group GID Attribute Type;`* `DN|UID`

Specifies the type of the attribute within the group object class. The attribute type can be either DN or UID.

-b, --auth_type FLAT BOB|FLAT SKIPNESTED|BOB SKIPNESTED

This attribute is a string that specifies the type of LDAP authentication mechanism to use for the given domain. AuthType can be either FLAT or BOB. FLAT means to use the existing one-level bind, while BOB indicates Bind_Search(Obtain)-Bind. In BOB authentication mode, AT uses a proxy account to bind with Active Directory. Then, AT searches for the distinguished name before it authenticates (binds) the user.

For RFC2307-compliant LDAP servers, you can disable nested group search and recursive group search for LDAP using the SKIPNESTED keyword.

-m, --admin_user *admin_user_DN*

This attribute is a string that contains the DN of the administrator user, and certain other users. The attribute can contain the DN of any user with search permissions for the user container or the user subtree that the UserBaseDN specifies. Configure this attribute to an empty string if the user container is searchable, and especially if users can search the user container anonymously. For example:

AdminUser=""

-w, --admin_user_password *admin_user_password*

This attribute is a string that contains the bind password of the user that is specified in AdminUser. If AdminUser is an empty string, this attribute must also be an empty string. For example, adminUserPassword="".

-p, --search_scope SUB|ONE|BASE

This attribute indicates the search scope. The search scope can be either SUB, ONE, or BASE.

addprpl -t root|ab|cluster|local -d *domain_name* -p *principal_name* [-s *password*] [-e *expiry_period_in_seconds*] [-q *default* | user | service] [-c] [-x] [-i] [-o] [[-b *host*[:*PBXPort*:*VxSSIOPServiceID*]]] [-y *domain_admins_domain_type* [:*domain_admins_domain_name*][-z *domain_admins_principal_name*]]

Creates authentication principals in the domain. You can only use this command when the broker is installed, and you are the root user.

-t, --pdrtype *PDR Type*

Specifies the type of private domain repository: Root broker, authentication broker, cluster, or local.

-d, --domain *Domain Name*

Specifies the name of the domain in which the principal is to be created.

`-p, --prplname` *Principal Name*

Specifies the name of the principal you want to create. The maximum length of the principal name is 64 characters.

`-s, --password` *Principal's Password*

Specifies the password for the new principal. The minimum password length is five characters.

`-e, --credexpiry` *Expiry Period in seconds*

Specifies the expiration interval in seconds. A hierarchy of intervals exists. If you set an expiry interval at the level of the individual principal, authentication uses the individual expiry interval. If the individual principal expiry is 0, authentication inherits the domain expiry. If the domain expiry is 0, it inherits the plug-in expiry. If the plug-in expiry is 0, it uses the global expiry.

`-q, --prpltype` *Principal Type*

Specifies the type of principal to create, whether a user or a service. Specify the principal type as `service` for a process. Specify the principal type as `user` for an individual user. The default principal type is `user`.

`-c, --can_proxy` *Can Proxy*

Indicates that the principal can act as proxy for another principal. This option is useful for Web server credentials. For example, it is useful in a situation where the Web server uses back end proxy services for users who access the Web browser.

`-x, --can_accept_proxy` *Can Accept Proxy*

Gives the entity the rights to accept proxies. This case is useful for the back-end services of a Web server. Before it hands out a product Web credential, the Web server checks whether the receiving peer has been cleared to accept the product Web credential. It also checks whether it can accept the proxy.

`-i, --is_broker_admin` *Is Broker Admin*

Gives the broker administrator privilege to the principal being created.

`-o, --is_domain_admin` *Is Domain Admin*

Gives the domain administrator privilege to the principal being created.

`-b, --broker` *BrokerName*`:`*PBXPort*`:`*VxSSIOPServiceID*

`-y, --domain_admin_domain` *Domain Admin's Domain*

-z, --domain_admin_prplname *Domain Admin's Principal Name*

authenticate [-d *domain_type:domain_name*][-p *principal_name* [-s *password*]] [-b *host*[{:*port*|*PBXPort*:*PBXServiceID*}]]

Use this command option to obtain a credential for an authentication principal from an authentication broker. A non-root user can run this command. You can run it even if only the client is installed.

-d, --domain *DomainType*:*DomainName*

Specifies the name and type of the domain that holds the principal. The private domain names do not need to be fully qualified ones. The given broker name without "*@fully_qualified_broker_name*>" is also accepted.

-p, --prplname *Principal Name*

Specifies the name of the principal that is to be authenticated. This argument is optional if you use "localhost" as the domain type. This argument is also optional if you use "nt" as the domain type and want to use SSPI. For other domain types, this argument is required.

-s, --password *Principal's password*

Specifies the password of the principal to authenticate. This argument is optional if you use "localhost" as the domain type. This argument is also optional if you use "nt" as the domain type and want to use SSPI. For other domain types, this argument is required.

-b, --broker *BrokerName*:*Port* (or) *BrokerName*:*PBXPort*:*PBXServiceID*

The host, port, and service ID of the broker. If a domain-broker mapping is already present, providing the broker information is optional.

Examples:

```
haat authenticate -d vx:broker -p TomSawyer
```

```
haat authenticate -d vx:broker -p Tom
```

changepasswd -t *root*|*ab*|*cluster* -d *domain_name* -p *principal_name* [-c *oldpasswd*] [-n *newpasswd*] [-r *repnewpasswd*]

Changes a password for a principal. The password is optionally provided on the command line. If not specified on the command line, it is prompted for in non-echo mode.

-t, --pdrtype *PDR Type*

Specifies the type of private domain repository: Root broker, authentication broker, or cluster.

-d, --domain *Domain Name*

Specifies the name of the primary domain.

`-p, --prplname` *`Principal Name`*

Specifies the name of the principal whose password is to be changed.

`-c, --currentpasswd` *`Current Password`*

Specifies the old password.

`-n, --newpasswd` *`New Password`*

Specifies the new password. The minimum acceptable password length is five characters.

`-r, --repeatednewpasswd` *`Repeat New Password`*

Specifies the new password, which you retype as confirmation.

For example:

```
haat changepasswd -t ab -d broker -p TomSawyer -c LetTomIn
-n PleaseLetTomIn -r PleaseLetTomIn
```

`checkclockskew -b` *`host`* `[-s yes]`

Checks the time on a system on which the Symantec Product Authentication Service is installed. Checks to see that the system time and GMT are within 75 minutes of one another. If there is a difference greater than 75 minutes, the installation returns an error.

`-b, --broker` *`broker host`*

Specifies the local system or the remote system.

`-s` This command option returns either 1 or 0. The return value 1 indicates failure, meaning that the clock skew has been detected. The return value 0 indicates success.

This command is used as follows:

Example 1:

```
haat checkclockskew -b mybroker.veritas.com
```

The output is one of the following:

- Clock skew detected between this machine and mybroker.veritas.com UMI error code
- No Clock Skew detected

Example 2:

```
haat checkclockskew -b mybroker.veritas.com -s
```

0

`createpd -t ab|cluster|local -d domain_name[-s domain_admin_password] [-c expiry_period_in_seconds] [[-b host[:PBXPort:VxSSIOPServiceID]]] [-x broker_admin_domain_type [:broker_admin_domain_name] [-a broker_admin_identify]`

Creates a private domain in the repository. The name must be unique.

In earlier versions, this command also created the default admin principal with a password of Vxadmin. Current implementation no longer creates that principal.

You can only use this command when the broker is installed, and you are a root user.

`-t, --pdrtype PDR Type`

Specifies the type of private domain repository, whether authentication broker or local. Root broker is not an option because you cannot create or delete domains in the root private domain repository. The root private domain repository has only one domain, where all authentication broker's identities are stored.

`-d, --domain DomainName`

Specifies the name of the domain to be created. The domain name cannot be more than 63 characters.

`-s, --domain_admin_password Domain Admin Password`

Specifies the domain administrator password for the domain being created. If not provided, the default admin account is not created.

`-c, --credexpiry Credential_Expiry`

`-b, --broker BrokerName:PBXPort:VxSSIPServiceID`

`-x, --broker_admin_domain Broker Admin Domain`

`-a, --broker_name_admin_prpl Broker Admin Principal Name`

`deleteauthsequence -d plugin_name`

Deletes a plug-in from the current authentication sequence. The plug-in may be anywhere in the auth sequence list.

`-d, --delete Plugin Name`

Specifies the name of the plug-in to be deleted.

`deletebrokerdomain -b host[{:port|:PBXPort:PBXServiceID}] -d domain_type:domain_name [-g]`

Deletes a mapping of a domain to a broker. Such a mapping indicates which broker the user should approach when the user tries to authenticate to a particular domain. You can specify whether this entry should be deleted from the local registry.

`-b, --broker BrokerName:Port (or) BrokerName:PBXPort:PBXServiceID`

Specifies the host, port, or service ID of the broker.

`-d, --domain DomainType:DomainName`

Specifies the name of the domain to delete.

`-g, --global Global Map`

Indicates that the entry should be removed from the local registry. For AT 6.0, all the entries are updated in the local registry.

Examples:

To delete the mapping of `nt:NewBrokerDomain` on `MyHost:14159:service_ID` from the configuration:

```
haat deletebrokerdomain -b MyHost:14159:service_ID -d nt:NewBrokerDomai
```

`deletecred -d domain_type:domain_name [-p principal_name [-b host[{:port|:PBXPort:PBXServiceID}]]]`

Deletes a credential from a store. Provide the user name and domain details. To delete the credential, use the same details you provided when you requested the credential.

`-d, --domain DomainType:DomainName`

Specifies the name of the domain that holds the principal whose credential is to be deleted.

`-p, --prplname PrincipalName`

Specifies the name of the principal whose credential you want to delete.

`-b, --broker BrokerName:Port (or)BrokerName:PBXPort:PBXServiceIDP`

Specifies the host, port, or service ID of the broker. Although port is specified here, it is ignored in the processing of this command. If the broker is specified, only the credential from a specific broker is deleted. There can be two different credentials for the same authentication principal from two different authentication brokers.

Example:

```
haat deletecred -d nt:NewDomainName -p TomSawyer
```

`deleteexpiredcreds`

Deletes expired credentials from a store.

`deleteexpiredsessions`

Deletes expired sessions.

`deletepd -t ab|cluster|local -d domain_name [-s]`

Deletes a private domain in the repository. Deleting a domain deletes the principals in the domain, along with the domain itself.

`-t, --pdrtype PDR Type`

Specifies the type of private domain repository: Authentication broker, cluster, or local.

Root broker is not an option for this command, because you cannot create or delete domains in the root private domain repository. The root private domain repository has only one domain, where all authentication broker's identities are stored.

`-d, --domain DomainName`

Specifies the name of the domain to be deleted.

`-s, --silent Silent Option`

Disables the confirmation messages.

`deleteprpl -t root|ab|cluster|local -d domain_name -p principal_name [-s]`

Deletes a principal from a private domain.

`-t, --pdrtype PDR Type`

Specifies the type of private domain repository: Authentication broker, cluster, or local.

`-d, --domain DomainName`

Specifies the name of the domain in which the principal resides.

`-p, --prplname Principal Name`

Specifies the name of the security principal.

`-s, --silent Silent Option`

Disables the confirmation messages.

Example:

```
haat deleteprpl -t ab -d broker -p TomSawyer
```

`exportrootcred -o root_credential_file`

To facilitate inter-operability with third-party services, the AT client must be able to:

- Import the third-party CA certificates into AT's trusted store
- Export the AT root credential in a standard format that third-party services can import.

After the root certificate and intermediary signing certificate are exchanged, both parties are in a position to establish communications.

Use this command option to export the trusted certificates of the AT into the file that is specified on the command line. All the files are exported in PEM format. If multiple certificates are present in the trusted store, they are all exported into the same file.

`-t, --out root credential file name`

Specifies the file that holds third-party root certificate(s) that are in PEM format.

`getbrokeruuid -b host[{:port|:PBXPort:PBXServiceID}]`

Gets the broker UUID.

`-b, --broker BrokerName:PBXPort:VxSSIPServiceID`

Specifies the host, port, or service ID of the broker.

`importrootcred -i 3rd_party_CA_cert_file`

To facilitate inter-operability with third-party services, the AT client must be able to:

- Import the third-party CA certificates into AT's trusted store
- Export the AT root credential in a standard format that third-party services can import.

After the root certificate and intermediary signing certificate are exchanged, both parties are in a position to establish communications.

Use this command option to import the trusted certificates that are in PEM format into the AT trusted store. Multiple PEM encoded certificates present in the same file are imported together. After they are imported, the certificates can be used to set up secure SSL sessions.

If you add duplicate root/CA certificates, the number of imported credentials increases, but only one copy is stored in the trusted store.

`-i, --3rd party CA certifiate file name`

Specifies the file that holds third-party root certificate(s) that are in PEM format.

`listldapdomains`

Lists all the LDAP domains in the authentication broker. This command needs no additional parameters.

`listpd -t root|ab|cluster|local [[-b host[:PBXPort:VxSSIOPServiceID]]] [-x broker_admin_domain_type [:broker_admin_domain_name] [-a broker_admin_identity]]`

Lists the domains inside the private domain repository of a local broker or a remote broker. To list the domains from a remote broker, first authenticate with the remote broker using the remote broker's broker admin identity.

`-t, --pdrtype PDR Type`

Specifies the type of private domain repository: Root broker, authentication broker, cluster, or local.

`-b, --broker BrokerName:Port(or) BrokerName:PBXPort:PBXServiceIDP`

Specifies the host, port, or service ID of the broker.

`-x, --broker_admin_domain`

Specifies the broker admin domain type and name.

`-a, --broker_admin_prplname`

Specifies the broker admin principal name.

`listpdprincipals -t root|ab|cluster -d domain_name`

Lists all the principals in the private domain.

`-t, --pdrtype PDR Type`

Specifies the type of private domain repository: Root broker, authentication broker, or cluster.

`-d, --domain Domain Name`

Specifies the name of the private domain whose principals you want to list.

`login -d domain_type[:domain_name] [-p principal_name] [-b host[{:port|:PBXPort:PBXServiceID}]]`

The `login` option is not the same as `authenticate`. The system requires you to authenticate before you can run `haat login`.

Use the `login` command option to set the context of the security principal that executes remote administration commands, such as `createpd`, `listpd`, and `addprpl`. On UNIX, the logon context is set per shell. Each shell has a

separate session (POSIX session, except on Linux). On Windows, after the security principal is logged on, its context applies to all of the remote `haat` commands that are executed on any shell. If you pass context as part of a remote administration command, the command-line context takes precedence over the logon context that is already set.

The logged on session eventually expires if it is not used.

`-d, --domain` *`DomainType`*`:`*`DomainName`*

Specifies the name of the domain that holds the security principal that executes remote commands.

`-p, --prplname` *`Principal Name`*

Specifies the name of the security principal that executes remote commands.

`-b, --broker` *`BrokerName`*`:`*`Port`* `(or)` *`BrokerName`*`:`*`PBXPort`*`:`*`PBXServiceIDP`*

Specifies the host, port, or service ID of the broker.

`logout`

Unsets/removes the security principal context for remote administration.

`pullbrokerattribs -b` *`host`*`[{:`*`fport`*`|:`*`PBXPort`*`:`*`PBXServiceID`*`}] [-v] [-p] [-i] [-c] [-n] [-m] [-r] [-f]`

Retrieves attributes from the authentication broker or root broker, on an authentication broker, or on a client system.

If you execute this command on a root broker, it retrieves the domain broker maps from the specified broker. Executing this command is useful when the root broker is unreachable and unable to push the stored broker's information. When the domain maps are pulled, they are stored in the regular domain maps section so that `showallbrokerdomains` reflects this new information from the specified broker.

Executed on a client-only system, this command option helps the client system retrieve the broker attributes, such as the cluster name and the broker version. Because it is client only, any pulled domain maps are displayed only, not stored.

`-b, --broker` *`BrokerName`*`:`*`Port`* `(or)` *`BrokerName`*`:`*`PBXPort`*`:`*`PBXServiceID`*

Specifies the host, port, or service ID of the broker.

`-v, --version` *`broker_version`*

Displays the version of the broker.

`-p, --port` *`broker_port`*

Specifies the port of the broker.

`-i, --ispbxenabled whether_PBX_is_enabled`

Enables the is_pbx_enabled flag of the broker.

`-c, --clustername broker_cluster_name`

Specifies the cluster name of the broker.

`-n, --name broker_name`

Specifies the broker name.

`-f, --fqhn brokers_fully_qualified_host_name`

Specifies the fully qualified host name of the broker.

`-m, --mode broker_mode`

Specifies the domain maps of the broker.

`-r, --registered products_registered`

Specifies the products that are registered with the broker.

`pushbrokerattribs -b host[{:fport|:PBXPort:PBXServiceID}]`

Pushes the All Domain-Broker Maps to all other authentication brokers that are registered with that root broker. You can perform a push under the following circumstances:

- Whenever an authentication broker gets added to a root broker
- Whenever an authentication broker gets deleted from a root broker
- Whenever a root broker pulls the All Domain-Broker Maps from a particular authentication broker at a fixed interval. The interval is defined in localconfig.

`-b, --broker BrokerName:Port`

(or)

`BrokerName:PBXPort:PBXServiceID`

Specifies the host, port, or service ID of the broker.

`refreshtrust`

Refreshes trust with the primary authentication server. Trust refresh parameters must have already been specified with the `settrustrefreshparams` command option.

`removeldapdomain -d domain_to_be_removed`

Removes an LDAP domain from the authentication broker.

`-d, --domain DomainName`

Specifies the symbolic name that uniquely identifies the LDAP domain.

`removesessioncache [-n session_cache_name] [-k] [-s]`

Removes the specified session cache files and optionally removes the cache configuration from the AT configuration. You can choose to remove only the cached sessions and keep the configuration intact.

`-n` Specifies the name of the session cache to be removed.

`-k` A flag indicating that cache configuration information should be retained.

`-s` Silent option.

`removetrust -b host[{:fport|:PBXPort:PBXServiceID}] [-n root_broker_name]`

Deletes the root certificate that comes from the mentioned broker.

`-b, --broker BrokerName:Port (or) BrokerName:PBXPort:PBXServiceID`

Specifies the host, port, or service ID of the broker.

`-n, --cname Root Broker Name`

Specifies the name of the root broker.

`renewcredential -d domain_type:domain_name -p principal_name -b host[{:fport|:PBXPort:PBXserviceID}]`

Renews the credential of a given principal, when you provide a domain and broker.

`-d, --domain DomainType:DomainName`

Specifies the name of the domain that holds the credential to be renewed. The command requires the vx domain type.

`-p, --prplname PrincipalName`

Specifies the name of the principal whose credential is to be renewed.

`-b, --broker BrokerName:Port (or) BrokerName:PBXPort:PBXServiceID`

Specifies the hot, port, or service ID of the broker.

`resetpasswd -t root|ab|cluster -d domain_name -p principal_name [-n newpasswd] [-r repnewpasswd]`

The administrator uses this command to reset a password when the authentication principal forgets the password.

The command does not require that you type the old password.

`-t, --pdrtype PDR Type`

Specifies the type of private domain repository: A root broker, authentication broker, or cluster.

`-d, --domain` *Domain Type*

Specifies the name of the primary domain.

`-p, --prplname` *Principal Name*

Specifies the name of the principal whose password is to be changed.

`-n, --newpasswd` *New Password*

Specifies the new password. The minimum acceptable password length is five characters.

`-r, --repeatednewpasswd` *Repeat New Password*

Specifies the new password, which you retype as confirmation.

`restorebroker [-a` *complete_path*`] [-s]`

Stops the AT service before anyone runs this command.

This command option restores the broker from the archived snapshot directory, if it contains the configuration that was last backed up by `haat showbackuplist`. The command option checks whether the snapshot directory is present. If it is present, `haat restorebroker` restores it back to the original position.

`-a, --archivedloc` *complete_path_to_the_snapshot_location*

Specifies the complete path of the archived material. If you use this option, the command ignores the location in the VRTSatlocal.conf file.

`-s, --silent`

Runs the command silently, without any prompt for restore. The default location is picked up from the `VRTSatlocal.conf` file.

`setbrokerlog -l 0|1|2|3|4`

Sets the broker log level.

`-l, --loglevel`

Sets the broker log level. The level is an integer between 0 and 4.

`setclockskewtolerance -t` *clock_skew_tolerance_in_seconds*

Sets the clock skew tolerance in seconds.

`-t, --tolerance` *clock skew tolerance in seconds*

Specifies the number of seconds that the credentials remain valid after expiry.

`setcredstore -t file|memory|registry -s file_if_file_type [-e]`

Sets credential store details. The details contain the store type (in memory, on file, or in the Windows registry). If it is on file, you can specify and see the file location.

`-t, StoreType`

Specifies the type of credential store for which you want to specify details. The store type may be `file`, `memory`, or `registry`.

`-s, StoreFileName`

Specifies the path where the file resides, if you have chosen `file` as the type of credential store.

`-e, Obfuscate`

Indicates that obfuscation is enabled.

`setdomaindiscoveryinterval -i interval_in_seconds`

Specifies, in seconds, how often the authentication broker discovers the domains that it supports. The default is 30 minutes. Use this command to change the interval to another value. You can turn off discovery by setting the value to 0. The authentication broker realizes, however, that you may change your mind about whether or not to discover. Therefore, if the value is 0, the broker refrains from discoveries, but it checks every 30 minutes to see whether you have changed your mind. If the value is set to *n* seconds, the broker does a domain discovery every *n* seconds. It also checks every *n* seconds to see whether you have turned off discovery or have changed your mind about how often to do it.

`-i, interval_in_seconds`

Specifies how often, in seconds, the authentication broker discovers the domains that it supports.

`setexpiryintervals -p plugin_name -t default|user|service|webcredential -e expiry_period`

Sets any of the levels of credential expiry: `default`, `user`, `service`, `webcredential`. These intervals are set at the plug-in level. To go up one level, set the expiry to 0 at that level. For example, you may want to go from a principal to a domain to a plug-in. If you want to remove the principal expiry and obtain a certificate that is based on the domain expiry, set the principal expiry to 0.

`-p, --pluginname Plugin Name`

Specifies the name of the plug-in where the credential expiry period is to be set.

`-t, --prpltype` *Principal Type*

Specifies the type of expiry to be set. For operating system domains or public domains, only the default expiry policy is used. Symantec Product Authentication Service cannot differentiate between a user account and a service account. Therefore, setting the `user` or `service` expiry policies for native domains may not have any effect on the actual credential expiry.

`-e, --credexpiry` *Credential Expiry*

Specifies the expiry period in seconds.

`setispbxexchflag [-e|-d]`

Sets the PBX Exchange Installed attribute to either enabled (`-e`) or disabled (`-d`). If you select enabled, a broker starts the PBX-related services. PBX-related services include PBX-based authentication support and remote administration.

`-e, --enable` *enable the PBX exchange flag*

`-d, --disable` *disable the PBX exchange flag*

`setloglevel -l 0|1|2|3|4 [-f` *Log_File_Name*`]`

Sets the log level. If you specify `-f`, the log level setting is applied to the client side.

`-l, --loglevel`

Specifies the log level. Client-side logging has five logging levels. By default, the log level is 0. For client-side logging, you can specify the name of the file to store the client-side log messages.

The server side has four logging levels. By default, the server-side log level is 1.

The following log levels exist:

Log level 0 does not log anything in the log files.

Log level 1 logs only critical error messages that require administrator attention.

Log level 2 logs all errors.

Log level 3 logs all errors and warnings.

Log level 4 logs everything, including trace messages.

`-f, --filename`

Specify the `-f` option for client-side logging and indicate the name of the file to store the client-side log messages. When the log file size reaches the

maximum, the file is moved to filename.1, filename.2, filename.3, filename.4, and filename.5.

`setmaxlogfiles -n Number_of_files(int)`

Specifies the maximum number of log files to preserve. After all the log files are filled, the oldest is recycled.

`-n, --numfiles Number_of_files`

Specifies the maximum number of log files to preserve.

`setmaxlogfilesize -s file_size_in_bytes`

Specifies the maximum size of the log files.

`-s, --size file_size_in_bytes`

Specifies the maximum file size in bytes.

`setpd -t root|ab|cluster|local -d domain_name -c expiry_period_in_sec`

Sets the attributes of the private domains. Currently the only attribute you can set using this command is the expiry period.

`-t, --pdrtype PDR Type`

Specifies the type of private domain repository: Root broker, authentication broker, cluster, or local.

`-d, --domainname Domain Name`

Specifies the name of the domain whose attributes are to be set.

`-c, --credexpiry Credential Expiry`

Specifies the expiry period in seconds.

`setpdr -t root|ab|cluster|local -f fqfn_of_pdr_file`

Changes the default location of the private domain repository. When the PDR file is changed, the current configuration is not immediately saved to the new PDR file. When you restart VCS One, the new PDR file is loaded.

`-t, --pdrtype PDR Type`

Specifies the type of private domain repository: Root broker, authentication broker, cluster, or local.

`-f, --pdrfile PDR File Name`

Specifies the fully qualified file name of the file that serves as the private domain repository. Enclose the path name in quotes if it contains a space.

`setsecuritylevel -l low|medium|high`

Sets the security level.

-l, --level *SecurityLevel*

Specifies the security level.

setsessioncacheparams {[-n *session_cache_name*] [-m *max_sessions*] [-u on|off] [-s 1|2|3]}

Configures the session cache and initializes the on-disk session cache database. You can also use this command option to turn the session cache that is already configured on or off.

-n, *session_cache_name*

Specifies the name of the session cache database. If you do not specify a name, a default name is used. Currently, this parameter is not used.

-m, *max_sessions*

Specifies the maximum number of sessions to hold in the on-disk session cache. When the session database already holds more sessions than specified, some of the sessions are dropped to reduce the size. By default, this value is 20*1024.

-u, on|off

Specifies whether the session cache is on or off.

-s, *Session cache storage type*

Specifies one of the following cache storage types:

1 In-memory cache only (default)

2 On-disk cache only

3 In-memory and on-disk cache

setsystemtrustdir {[-u on|off] [-t *directory*]}

Sets the system trust directory.

-u, --usetrustdir on|off

Indicates whether the trust directory is on or off.

-t, --trustdir *DirectoryName*

Specifies the name of the directory that is used as the system trust directory.

setuptrust -b *host*[{*port*|:*PBXPort*:*PBXServiceID*}] -s low|medium|high [-f *filename*| -r *root_hash_in_hex*]

Use this command to:

- Contact the broker to be trusted.
- Obtain its certificate or details over the wire.

- Add to the trust repository if the furnished details are trustworthy. A non-root user can run this command. You can run it even if only the client is installed.

`-b, --broker` *`BrokerName`*`:`*`Port`* `(or)` *`BrokerName`*`:`*`PBXPort`*`:`*`PBXServiceID`*

Specifies the host, port, or Service ID of the broker to be trusted.

`-s, --securitylevel` *`SecurityLevel`*

Specifies the level of security that you want to set.

`-f, --hashfile` *`HashFileName`*

Specifies a binary file containing the root hash. Trust is set up in high security mode. Setup trust fails if the supplied root hash does not verify.

`-r, --hash` *`HashString`*

Specifies the root hash in hexadecimal format. Trust is set up in high security mode. Setup trust fails if the supplied root hash does not verify.

`settrustrefreshparams {-b` *`host`*`[{:`*`port`*`|:`*`PBXPort`*`:`*`PBXServiceID`*`}] [-a yes|no] [-t` *`refresh_interval`*`]}]`

Stores the trust refresh parameters for a given authentication server.

`-b, --broker` *`BrokerName`*`:`*`Port`* `(or)` *`BrokerName`*`:`*`PBXPort`*`:`*`PBXServiceID`*

Specifies the authentication server name, which can be a host name or an IP address. Additionally, you can also specify a port number or a PBX service ID. If the specified value is a number, it is treated as a port number. If it is not a number, it is treated as a PBX service ID. You need to specify at least one of the parameters.

`-a yes|no`

Specifies the auto trust refresh option. If `yes`, the vrtsAtSecconnConnect() and vrtsAtSecConnAccept() APIs attempt a trust refresh whenever they come across an unknown root credential. The default value is `no`.

`-t` *`refreshinterval`*

Specifies the auto trust refresh interval in seconds. The default is 1800 seconds.

`showallbrokerdomains [-g]`

Displays all the mappings of domain to broker. Results show the broker name, the broker port, the domain name, and the domain type.

Domain maps indicate what broker and port to approach to authenticate a given domain of a given type. The global option indicates whether this

mapping is for all principals or for the current operating system logged-on principal.

`-g, --global Global Map`

Shows the information for the local registry. All the entries are updated to the local registry.

`showalltrustedcreds`

Displays a list of all trusted credentials (that is, root certificates). The UUID from the credential is also displayed.

`Showauthsequence`

Displays the current chain of authentication plug-ins.

`showbackuplist [-f file_name]`

Use this command option to:

- List critical files and directories to back up
- List the names of the backed-up files, if the names differ from the original names
- List registry keys to back up
- Back up the displayed list of files

`-f, --filename FileName`

`showbrokerhash`

Displays the root broker hash. The root broker administrator publishes the root broker hash so that users can set up trusts. Publishing is done using their company's accepted security-related information dissemination tools.

`showbrokermode [-t]`

This command option only works if you are an administrator or superuser. Use it to display the current mode of the broker on the system where you run this command option. This command option outputs one of the following values:

0: The broker is not configured yet.

1: The broker is running as an authentication broker only.

2: The broker is running as root broker only.

3: The broker is running as root + authentication broker.

`-t, --text display the broker mode in text`

Displays the broker mode in text.

`showbrokers -d domain_type:domain_name`

Displays the brokers for a particular domain.

`-d, --domain DomainType:DomainName`

Specifies the domain for which the brokers are to be displayed.

`showbrokertag -a|-r`

Displays the broker tag. The broker tag is the default domain suffix for all the private domains. Unless you override it with the `setbrokertag` command, the tag is the same as the fully qualified host name.

`-a` Shows the broker tag that the authentication broker uses. If the tag is not present (that is, the broker is not yet configured), the output states that the authentication broker tag is not present.

`-r` Shows the broker tag that the root broker uses. If the tag is not present (that is, the broker is not yet configured), the output states that the root broker tag is not present.

`showclockskewtolerance`

Shows the current clock skew tolerance. Clock skew tolerance is a variable that specifies the number of seconds that the credentials remain valid after the expiry.

`showcred [-d domain_type:domain_name [-p principal_name [-b host[{:port|:PBXPORT:PBXServiceID]]]]]`

Displays the credentials that are available in the local repository. Use options to filter the search. If you run this command without options, it returns all credentials for the same authentication principal from different authentication brokers. If you do not provide broker information, the command shows all the credentials that belong to the authentication principal.

The UUID from the credential is also displayed. A non-root user can run this command. You can run it even if only the client is installed.

`-d domain_name:domain_type`

Specifies the name of the domain that holds the principal whose credentials you want to display.

`-p principal_name`

Specifies the name of the principal whose credentials you want to display.

`-b, --broker BrokerName:Port (or) BrokerName:PBXPort:PBXServiceID`

Specifies the host, port, or service ID of the broker.

`showcredinfo -t identity_tag [-e]`

Displays the principal information and domain information of a remotely provisioned identity on the target system.

`-t identity_tag`

Specifies the unqualified identity tag. When a unique identity is provisioned on a large number of systems, the complete principal name is the tag.

`-e` Displays the identify information in English.

`showcredstore`

Displays credential store details. These details contain the store type (in memory, on file, in the registry, etc.). If the store type is on file, the file location is displayed. The UUID from the credential is also displayed.

`showdomaindiscoveryinterval`

Indicates how often the authentication broker discovers the domains that it supports. You can change that interval with the `haat setdomaindiscoveryinterval` command.

You can turn off discovery by setting the value to 0. However, the authentication broker continues to listen, in case you turn discovery back on. If you set the value to 0, the broker does not do a discovery. Instead, it checks every 30 minutes to see if you have a discovery or have turned discovery back on.

The process is a discovery event that is scheduled every *n* seconds. During the discovery event, the current interval time is checked first. If the current interval time is 0, the discovery is skipped and is scheduled to occur in 30 minutes. If the current interval time is not 0, discovery occurs and the next discovery is scheduled to occur in *n* seconds.

`showdomains -p plugin_name`

Displays the domains that the specified plug-in supports.

`-p, --pluginname Plugin Name`

Specifies the name of the plug-in whose supported domains you want to see.

`showexpiryintervals -p plugin_name`

Displays the intervals of the credential expiry that have been set. This command option displays one of four levels of credential expiry types: Generic, user, Web, and service principal expiry intervals. These intervals are set at the plug-in level. The private domain supports a generic expiry interval.

`-p, --pluginname Plugin Name`

Specifies the name of the plug-in whose credential expiry levels you want to see.

`showglobalplugininfo`

Shows the credential expiry policies for all plug-ins. The order in which credential expiry policy is applied is:

1. Individual principal expiry policy
2. Domain expiry policy
3. Plug-in expiry policy
4. Global, all plug-ins expiry policy

`showispbxexchflag`

Shows if the PBX Exchange Installed flag is set on the broker. The output is 1 if the flag is set and 0 if it is not. If the flag is set, the broker uses PBX-related services, such as PBX-based authentication support and remote administration.

`showpd -t root|ab|cluster|local -d` *`domain_name`*

Displays the attributes of the private domains. Currently, the command displays only the expiry period.

`-t, --pdrtype` *`PDR Type`*

Specifies the type of private domain repository: Root broker, authentication broker, cluster, or local.

`-d, --domainname` *`Domain Name`*

Specifies the name of the domain whose attributes you want to see.

`showpdr [-t root|ab|cluster|local]`

Displays the locations of the private domain repositories.

`-t, --pdrtype` *`PDR Type`*

Specifies the private domain repository type: Root broker, authentication broker, or cluster.

`showplugininfo -p` *`plugin_name`*

Shows the plug-in details; the plug-in name, the expiry period, the maximum user name length, and how many domains exist (including their names and types). This command option also indicates the case sensitivity of the user domain: `1` means case sensitive and `0` means case-insensitive.

`-p, --pluginname` *`Plugin Name`*

Specifies the name of the plug-in for which you want to see details. Plug-in names are vx, ldap, nis, nisplus, pam, and unixpwd.

`showprpl -t root|ab|cluster|local -d domain_name -p principal_name`

Displays the attributes of a principal, such as the principal type and the expiry policy, within a domain.

`-t, --pdrtype PDR Type`

Specifies the type of private domain repository: Root broker, authentication broker, cluster, or local.

`-d, --domainname Domain Name`

Specifies the name of the domain in which the principal resides.

`-p, --prplname Principal Name`

Specifies the name of the principal whose attributes you want to see.

`showrootbroker`

Displays the root broker for which the system is configured.

`showsecuritylevel`

Displays the security level.

`show sessioncacheparams`

Displays the existing session cache. The parameters include the maximum on-disk size and information about whether or not the cache is in use.

`showsystemtrustdir`

Displays whether the systems-wide trust information is in use and the corresponding directory. This command option displays the system default trust directories as a colon-separated list. This directory is platform specific and supported by OpenSSL. The command displays whatever OpenSSL picks and the directory value that is stored in the SSL_CERT_DIR environment variable. All the root certificates in those directories are for trusted roots. The command may appear to return a directory that does not exist.

`showtrustrefreshparams`

Returns the trust refresh parameters for the primary authentication server. Output includes the authentication server information (host name, port, or PBX service ID), auto refresh flag, and the refresh interval.

`showversion`

Displays the version of the Symantec Product Authentication Service command line interface. A non-root user can run this command. You can run it even if only the client is installed.

`updateplugin -p plugin_name -a attribute_name -v attribute_value -t int|string`

Updates the plug-in information. This command option works with all the plug-ins. You can use it to enable or disable a plug-in or to update any of the plug-in's attributes.

`-p, --pluginname Plugin Name`

Specifies the name of the plug-in to be updated.

`-a, --attrib_name Attribute Name`

Specifies the name of the attribute to be changed.

`-v, --value Attribute Value`

Specifies the new value of the attribute.

`-t, --type Attribute Type(int or string)`

Specifies the type of attribute. It can be either an integer or a string.

`updateprpl -t root|ab|cluster|local -d domain_name -p principal_name -q default|user|service -e expiry_period_in_sec [-x] [-y] [-i] [-o]`

Updates the attributes of the principal. In addition, you can turn on the "Is Broker Admin" or the "Is Domain Admin" attribute for the principal. By default, these attributes are off.

`-t, --pdrtype PDR Type`

Specifies the type of private domain repository: Root broker, authentication broker, cluster, or local.

`-d, --domain Domain Name`

Specifies the name of the domain in which the principal resides.

`-p, --prplname Principal Name`

Specifies the name of the principal whose attributes are to be updated.

`-q, --prpltype Principal Type`

Updates the principal type.

`-e, --credexpiry Credential Expiry Period in seconds`

The expiry period in seconds. To turn off the expiry period for a principal, set it to 0.

`-x --can_proxy Can Proxy`

Indicates that the principal can act as a proxy for another principal. This option is useful for Web server credentials where the Web server must proxy to its back-end services for a user using the Web browser.

`-y --can_accept_proxy` *Can Accept Proxy*

Gives the entity the right to accept proxies. This option is useful for the back-end services of a Web server. Before it hands out the user's product Web credential, the Web server checks whether:

- The receiving peer has been cleared to accept the product Web credential
- The receiving peer can accept the proxy

`-i --is_broker_admin` *Is Broker Admin*

The presence of the parameter `is_broker_admin` sets the security principal to be the broker admin. The absence of the parameter resets it. To verify the setting, use `haat showprpl`.

`-o --is_domain_admin` *Is Domain Admin*

The presence of the parameter `-o` sets the security principal to be the domain admin. The absence of the parameter resets it. To verify the setting, use `haat showprpl`.

`validategroup -g` *group_name* `[-d` *domain_type*`:`*domain_name* `-b` *host* `[{:`*port*`|:`*PBXPort*`:`*PBXServiceID*`}]]`

Checks the validity of a given group when you provide the name of the domain and the broker.

`-g, --groupname` *GroupName*

Specifies the name of the group to be validated.

`-d, --domain` *DomainType*`:`*DomainName*

Specifies the name of the domain that holds the group to be validated.

`-b, --broker` *BrokerName*`:`*Port* (or) *BrokerName*`:`*PBXPort*`:`*PBXServiceID*

Specifies the host, port, and service ID of the broker.

`validateprpl -p` *principal_name* `[-d` *domain_type*`:`*domain_name* `-b` *host* `[{:`*port*`|:`*PBXPort*`:`*PBXServiceID*`}]]`

Checks the validity of a given principal when you provide the name of the domain and broker.

`-d, --domain` *DomainType*`:`*DomainName*

Specifies the name of the domain that holds the principal to be validated.

`-p, --prplname` *Principal Name*

Specifies the name of the principal to be validated.

`-b, --broker` *`BrokerName`*`:`*`Port`* `(or)` *`BrokerName`*`:`*`PBXPort`*`:`*`PBXServiceID`*

Specifies the host, port, and service ID of the broker.

`whoami`

Use this command to view the current security principal context that was used to log on . The output is as follows:

domaintype:*domainname*:*prplname*:*host*:*port*

SEE ALSO

`haldapconf`(1M)

haattr

`haattr` – use to define new attributes, change default values, delete the attributes that are associated with resource types or vtypes, or display attributes and their values for cluster objects

SYNOPSIS

```
UNIX: /opt/VRTSvcsone/bin/hattr

Windows:  %VCSONE_HOME%\bin\haattr

haattr -add [-static|-temp] [-insensitive] type attribute [VALUETYPE]
[DIMENSION] [defaultvalue] [-platform platform] [-user user@domain
-domaintype domaintype]
haattr -add [-static|-temp] [-insensitive] -vtype vtype attribute
[VALUETYPE] [DIMENSION] [defaultvalue] [-user user@domain -domaintype
domaintype]
haattr -delete [-static|-temp] type attribute [-platform platform]
[-user user@domain -domaintype domaintype]
haattr -delete [-static|-temp] -vtype vtype attribute [-user
user@domain -domaintype domaintype]
haattr -default type attribute defaultvalue [-platform platform]
[-user user@domain -domaintype domaintype]
haattr -default -vtype vtype attribute defaultvalue [-user user@domain
-domaintype domaintype]
haattr -display {cluster | remotecluster | group | csg | system |
user | role} [-user user@domain -domaintype domaintype]
haattr -display {pframe|vframe} -vtype vtype [-user user@domain
-domaintype domaintype]
haattr -display {type [-platform platform] | -vtype vtype} [-user
user@domain -domaintype domaintype]
haattr -setproperty type attribute [-platform platform] {propertykey
propertyvalue} ... [-user user@domain -domaintype domaintype]
haattr -setproperty -vtype vtype attribute {propertykey propertyvalue}
... [-user user@domain -domaintype domaintype]
haattr -getproperty type attribute [-platform platform] [-user
user@domain -domaintype domaintype]
haattr -getproperty -vtype vtype attribute [-user user@domain
-domaintype domaintype]
haattr [-help]
```

```
haattr -version
```

AVAILABILITY

VRTSvcsonec

DESCRIPTION

The `haattr` command adds attribute metadata; that is, it adds the name, the *VALUETYPE*, the *DIMENSION*, and the default value for the attribute.

Use the `-add` option to add attributes to resource types or vtypes. By default, the values of attributes apply to objects on all nodes and are global in scope. Local resource attributes are those whose values can be defined to apply for a specific system.

Attributes may be static or temporary:

- Static attributes have predefined default values and apply to all resources of a specific type or all frames of a specific vtype.
- A temporary attribute for a resource type or a vtype serves a temporary purpose. Temporary attributes exist in memory, and you can add, modify, or delete them only when the VCS One engine is running. Temporary attributes are lost when the engine stops.

The *VALUETYPE* of an attribute may be one of the following:

- `string`: a string of characters. You specify the string using the `-string` option
- `integer`: an integer that you specify using the `-integer` option
- `boolean`: a Boolean, specified by the `-boolean` option

By default, *VALUETYPE* is a string.

The *defaultvalue* for an attribute is the initial value that all instances of that attribute have.

The *DIMENSION* of an attribute may be one of the following:

- `scalar`: a single value that is a string, integer, or Boolean--specified by the `-scalar` option
- `vector`: an ordered list of non-unique values. A vector can be a string or an integer. You specify the scalar value using the `-vector` option
- `keylist`: an unordered list of unique string values that you specify using the `-keylist` option

- `assoc:` an unordered list of name-value pairs, where the value is a unique string associated with an integer that you specify using the `-assoc` option

By default, *DIMENSION* is scalar.

The defaultvalue for an attribute is the initial value that all instances of the attribute have. Use the `hatype` command or the `havtype` command with the `-modify` option to change the values of static attributes without modifying the metadata.

For the `-platform` option, supported values for *platform* are:

- aix
- aix/rs6000 (alias aix)
- esx
- hpux
- Linux
- linux/x86 (alias Linux)
- solaris
- solaris/x86
- solaris/sparc (alias solaris)
- windows
- windows/x86

Use the explicit platform name where no alias is defined. When *platform* appears in any displays, the full platform name (not the alias) is shown.

A non-root user who has not run the `halogin` command can execute the `haattr` command using the `-user` *user@domain* option. This option runs the command with the privileges of the specified user. When you issue the command, enter your fully qualified domain user name and supply a password when prompted. If necessary, the `-domaintype` option can specify the type of domain against which the user is to be authenticated. Supported domain types include:

- "unixpwd"
- "nt"
- "nis"
- "nisplus"
- "ldap"
- "pam"

- "vx" (Symantec Private Domain)

The default domain type is "vx". The domain type is case sensitive.

When using `domaintype=unixpwd`, provide the system name as the domain portion. The domain must be a fully-qualified domain name (for example, sun01.engba.veritas.com).

See NOTES for how to specify "-" and "%" characters in the command line.

OPTIONS

`-add [-static|-temp] [-insensitive] type attribute [VALUETYPE] [DIMENSION] [defaultvalue] [-platform platform] [-user user@domain -domaintype domaintype]`

Add an attribute to the configuration for the specified resource type. All new resources and existing resources of the specified type are instantiated with this attribute and its default value.

You can modify the attributes of individual instances using the `hares` command.

Use the `-static` option to add a static attribute to the VCS One configuration for the specified resource type. The *defaultvalue* is stored in the type class, and has the same value for every resource of that type. When new types are instantiated, they are instantiated with static attributes.

You may modify static attribute values with the `hatype` command for resources. You cannot modify non-static attribute values.

Use the `-temp` option to add a temporary attribute to the VCS One configuration for the specified resource type. The *VALUETYPE* may be either a string (`-string`, the default), integer (`-integer`), or Boolean (`-boolean`).

You may define a temporary attribute while the VCS One engine is running.

By default, a newly added attribute is case sensitive. If you want to add an attribute and make it case insensitive, use the `-insensitive` option when adding it.

`-add [-static|-temp] [-insensitive] -vtype vtype attribute [VALUETYPE] [DIMENSION] [defaultvalue] [-user user@domain -domaintype domaintype]`

Add an attribute to the configuration for the specified vtype. All new frames and existing frames of the specified vtype are instantiated with this attribute and its default value.

You can modify the attributes of individual instances using the `haframe` command.

Use the -static option to add a static attribute to the VCS One configuration for the specified vtype. The *default* value is stored in the vtype class, and has the same value for every frame of that vtype. When new vtypes are instantiated, they are instantiated with static attributes.

You may modify individual attribute values with the havtype command for frames.

Use the -temp option to add a temporary attribute to the VCS One configuration for the specified vtype. The *VALUETYPE* may be either a string (-string, the default), integer (-integer), or Boolean (-boolean).

You may define a temporary attribute while the VCS One engine is running.

By default, a newly added attribute is case sensitive. If you want to add an attribute and make it case insensitive, you may do so using the -insensitive option when adding the attribute.

-delete [-static|-temp] *type attribute* [-platform *platform*] [-user *user@domain* -domaintype *domaintype*]

Delete attributes for the specified resource type and delete the attributes for all existing instances of the resource type.

-delete [-static|-temp] -vtype *vtype attribute* [-user *user@domain* -domaintype *domaintype*]

Delete attributes for the specified frame vtype and for all existing instances of the frame vtype.

-default *type attribute defaultvalue* [-platform *platform*] [-user *user@domain* -domaintype *domaintype*]

Change the default value for a non-static attribute of the specified resource type. Instantiate subsequent resource type instances with the new default value.

-default -vtype *vtype attribute defaultvalue* [-user *user@domain* -domaintype *domaintype*]

Change the default value for an attribute of the specified frame vtype and instantiate the subsequent instances of the frame vtype with the new default value.

-display {*cluster* | *remotecluster* | *group* | *csg* | *system* | *user* | *role*} [-user *user@domain* -domaintype *domaintype*]

For a specified object, display its attributes and include the name, *VALUETYPE*, *DIMENSION*, and default value (if any).

`-display {pframe | vframe} -vtype vtype [-user user@domain -domaintype domaintype]`

For a specified vtype of pframe or vframe, display its attributes and include the name, *VALUETYPE*, *DIMENSION*, and default value (if any).

`-display {type [-platform platform] | -vtype vtype} [-user user@domain -domaintype domaintype]`

For a specified resource type or vtype, display its attributes and include the name, *VALUETYPE*, *DIMENSION*, and default value (if any).

`-setproperty type attribute [-platform platform] {propertykey propertyvalue}… [-user user@domain -domaintype domaintype]`

Set or update the values of attribute properties for a specified resource type. You can modify only the properties of resource or frame attributes with the `-setproperty` option. You cannot modify the Type attribute properties.

`-setproperty -vtype vtype attribute {propertykey propertyvalue}… [-user user@domain -domaintype domaintype]`

Set or update the values of attribute properties for a specified frame vtype. You can modify only the properties of resource or frame attributes with the `-setproperty` option. You cannot modify Vtype attribute properties.

`-getproperty type attribute [-platform platform] [-user user@domain -domaintype domaintype]`

Displays the values of attribute properties for a specified resource type.

`-getproperty -vtype vtype attribute [-user user@domain -domaintype domaintype]`

Displays the values of attribute properties for a specified resource vtype.

`[-help]`

Display command syntax. When you enter the command and an option without arguments, syntax for the specific option displays.

`-version`

Display command version.

EXAMPLES

To display the usage syntax for a specific command option, enter the command and an option without arguments. For example, enter:

```
# haattr -add
```

To add a new user permissions attribute, `Permissions` for a FileOnOff resource, enter:

```
# haattr -add FileOnOff Permissions -assoc root rwx
```

For the default platform only, this command adds the attribute `Permissions` to all resources (current and future) of type FileOnOff, which is of the specified association *DIMENSION*. The attribute has the *VALUETYPE* "string" by default. The default value for all new instantiations and existing instantiations of FileOnOff resources is the name-value association `root rwx`.

To add a temporary attribute `SocketPortNumber` to the `Process` resource type definition, enter:

```
# haattr -add -temp Process SocketPortNumber -integer -scalar 0
```

For the default platform only, this command adds the temporary attribute `SocketPortNumber` to all resources of the type `Process`. The command continues adding the `SocketPortNumber` attribute for as long as the VCS One engine is running. The *VALUETYPE* is an integer and *DIMENSION* is a scalar. The default value for the SocketPortNumber for all instantiations of Process resources is 0.

In the following example, the default value of the Permissions attribute is changed for the FileOnOff resource:

```
# haattr -default FileOnOff Permissions root rwx user rw
```

In the following example, entering the command:

```
# haattr -getproperty FileOnOff Permissions
```

retrieves the following properties of the Permissions attribute for the FileOnOff resource:

#Property	Value
static	OFF
non_persistent	OFF
no_modify	OFF
no_run_modify	OFF
il8n	OFF
no_override	OFF
no_print	OFF
cteam	OFF
no_snap	OFF

local	OFF
local_parallel	OFF
scope	OFF
no_local	OFF
no_dump	OFF
temp	OFF
deprecated	OFF
obsolete	OFF
no_cfnum_update	OFF
important	OFF
must_configure	OFF
agent_encrypt	OFF
unique	OFF
propagate_proxy	OFF
propagate_group	OFF
target_resource	OFF
non_empty	OFF
lic_standard	OFF
description validation	0

In the following example, the property of the Permissions attribute is set:

```
# haattr -setproperty FileOnOff Permissions no_modify ON
```

In the following example, the Permissions attribute is deleted for the FileOnOff resource:

```
# haattr -delete FileOnOff Permissions
```

NOTES

You cannot use this command to modify the attributes that the system defined.

You cannot add attributes to the cluster, system, group, user, or role objects.

When you use the command to specify or modify an attribute value that begins with a dash ("-"), precede the value with a percent sign ("%"). For example, specify **-y** as **%-y**. Likewise, precede a value that starts with a percent sign with another percent sign.

SEE ALSO

`hares`(1M), `hatype`(1M), `haframe`(1M), `havtype`(1M), `halogin`(1M)

haclus

`haclus` – display and manage cluster attributes and their values

SYNOPSIS

```
UNIX: /opt/VRTSvcsone/bin/haclus

Windows: %VCSONE_HOME%\bin\haclus

haclus -add cluster [-user user@domain -domaintype domaintype]
haclus -delete cluster [-user user@domain -domaintype domaintype]
haclus -declare
haclus -display [cluster][-attribute attribute(s)] [-user user@domain
-domaintype domaintype]
haclus -list [-user user@domain -domaintype domaintype]
haclus -state [-user user@domain -domaintype domaintype]
haclus -value attribute [-clus cluster] [-user user@domain -domaintype
domaintype]
haclus -wait attribute attr_value [-time seconds] [-clus cluster]
[-user user@domain -domaintype domaintype]
haclus -modify modify_options
haclus [-help [-modify]]
haclus -version
```

AVAILABILITY

VRTSvcsonec

DESCRIPTION

You can use the `haclus` command to display cluster attributes and values.

A non-root user who has not run the `halogin` command can execute the `haclus` command using the `-user` ***user@domain*** option. This option executes the command with the privileges of the specified user. When you issue the command, enter your fully qualified domain user name and supply a password when prompted. If necessary, the `-domaintype` option can specify the type of domain against which the user is to be authenticated. Supported domain types include:

- "unixpwd"
- "nt"

- "nis"
- "nisplus"
- "ldap"
- "pam"
- "vx" (Symantec Private Domain)

The default domain type is "vx". The domain type is case sensitive.

See NOTES for how to specify "-" and "%" characters in the command line.

OPTIONS

`-add` *`cluster`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Add a remote cluster with the specified cluster name. This option applies in a VCS One global cluster environment. This command requires that you have the Add Cluster privilege at the VCS One cluster level to add a remote cluster.

`-delete` *`cluster`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Delete a remote cluster with the specified cluster name. This option applies in a VCS One global cluster environment. This command requires that you have the Delete Cluster privilege at the VCS One cluster level to delete a remote cluster.

`-declare`

`-display [`*`cluster`*`] [-attribute` *`attribute(s)`* `-user` *`user@domain`*
`-domaintype` *`domaintype`*`]`

Display the values of all cluster attributes or specified attributes for the specified cluster. If you do not specify a cluster, the command displays the attribute values for the local cluster.

`-list [-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Display a list of clusters that belong to a VCS One global cluster. The local cluster is indicated with an asterisk after its name.

`-state [-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Return the current state of the local and the remote clusters as seen from the local cluster. The local cluster is indicated with an asterisk after its name and the state of the local cluster is always listed first.

In addition to the cluster state, this option also displays the consolidated status of the network links for the remote clusters. See EXAMPLES.

`-value` *attribute* [`-clus` *cluster*] [`-user` *user@domain* `-domaintype` *domaintype*]

Display the value of a specified attribute.

In a global VCS One cluster, the `-clus` option displays the value of the attribute for the specified cluster. If you do not use the `-clus` option, the command returns the value of the specified attribute for the local cluster.

Use the `-value` option instead of the `-display` option to see a specific attribute value rather than a table of many attribute values.

`-wait` *attribute attr_value* [`-time` *seconds*] [`-clus` *cluster*] [`-user` *user@domain* `-domaintype` *domaintype*]

In a script, `-wait` directs the `haclus` command to wait until an attribute value changes as specified, or until the number of seconds specified by *seconds* is reached. The *seconds* variable is an integer specifying seconds. If *seconds* is not specified, `haclus` waits indefinitely.

The `-wait` option can be used only with changes to scalar attributes.

In a global VCS One cluster, use the `-clus` option to apply the `-wait` option to a remote cluster. If you do not use the `-clus` option, the `-wait` option is used for the specified attributes in the local cluster.

The scalar cluster-level attributes on the remote cluster are limited to those that are displayed using the `haclus -display` *remote_cluster* command.

See EXAMPLES.

`-modify` *-modify_options*

The `-modify` option lets you modify the values of some of the cluster's attributes. Some attributes are internal to VCS One and cannot be modified. You can modify any attribute that can be configured in main.xml.

Modifiable attributes can be of any type or dimension. Modifying some attributes may have subtle implications. See the *Veritas Cluster Server One User's Guide* for details about individual attributes.

Use the `-clus` option to specify the remote cluster whose attributes you want to modify.

`SCALAR`

```
haclus -modify attribute value [-clus cluster] [-user
user@domain -domaintype domaintype]
```

`VECTOR`

Use the following command only when the attribute has no value:

```
haclus -modify  attribute value [-clus cluster] [-user
user@domain -domaintype domaintype]
```

Only the following operations are allowed on vector attributes with defined values:

```
haclus -modify attribute -add key [-clus cluster] [-user
user@domain -domaintype domaintype]
```

```
haclus -modify attribute -delete keys [-clus cluster] [-user
user@domain -domaintype domaintype]
```

Note: You cannot delete an individual element of a VECTOR.

KEYLIST

Use the following command only when the attribute has no value:

```
haclus -modify attribute key [-clus cluster] [-user
user@domain -domaintype domaintype]
```

Only the following operations are allowed on keylist attributes with defined values:

```
haclus -modify attribute -add key [-clus cluster] [-user
user@domain -domaintype domaintype]
```

```
haclus -modify attribute -delete key [-clus cluster] [-user
user@domain -domaintype domaintype]
```

```
haclus -modify attribute -delete keys [-clus cluster] [-user
user@domain -domaintype domaintype]
```

ASSOCIATION

Use the following command only when the attribute has no value:

```
haclus -modify attribute {key value} [-clus cluster] [-user
user@domain -domaintype domaintype]
```

Only the following operations are allowed on association attributes with defined values:

```
haclus -modify attribute -add {key value} [-clus cluster]
[-user user@domain -domaintype domaintype]
```

```
haclus -modify attribute -update {key value} [-clus cluster]
[-user user@domain -domaintype domaintype]
```

```
haclus -modify attribute -delete key [-clus cluster] [-user
user@domain -domaintype domaintype]
```

```
haclus -modify attribute -delete -keys [-clus cluster] [-user
user@domain -domaintype domaintype]
```

`-help [-modify]`

This option prints the command syntax. If the `-modify` option is specified, it prints the usage message for modifying the values of attributes. When you enter the command and an option without arguments, syntax for the specific option displays.

`-version`

Display the version for the command.

EXAMPLES

To display the usage syntax for a specific command option, enter the command and an option without arguments. For example, enter:

```
# haclus -value
```

To use the `-wait` option in a script to direct the `haclus` command to wait until the cluster changes to a `RUNNING` state, enter:

```
# haclus -wait ClusterState RUNNING
```

To display the state of the clusters in a global VCS One cluster, run the following command:

```
# haclus -state
```

ClusterName	ClusterState
c1*	RUNNING
c2	RUNNING | LINK_UP

NOTES

When using the command to specify or modify an attribute value that begins with a dash ("-"), precede the value with a percent sign ("%"). For example, specify `-y` as `%-y`. Likewise, precede a value that starts with a percent sign with another percent sign.

SEE ALSO

`halogin`(1M), `hacsg`(1M)

haconf

haconf – manage VCS One configuration

SYNOPSIS

```
UNIX: /opt/VRTSvcsone/bin/haconf

Windows: %VCSONE_HOME%\bin\haconf

haconf -cleandb
haconf -loaddb [-force] [xml_dir]
haconf -cftoxml cf_dir xml_dir -platform default_platform
haconf -dbtoxml [-force] xml_dir
haconf -dbtocmd cmd_dir
haconf -xmltocmd xml_dir|xml_file cmd_dir
haconf -verify [xml_dir|-db]
haconf -version
haconf -dbstatus
haconf -help
The default directory is:
UNIX and Linux: /etc/VRTSvcsone/conf/confxml
Windows Simulator: C:\Program Files\Veritas\Cluster Server One
Simulator\conf\confxml
Windows: C:\Program Files\Veritas\Cluster Server One\conf\confxml
```

AVAILABILITY

VRTSvcsonec

DESCRIPTION

The haconf utility is provided for managing the VCS One configuration. The utility can do the following:

- Read the configuration files in the CF format (VCS configuration) and convert the files to the XML.
- Read XML configuration files and populate database configuration tables.
- Dump the database configuration to XML files.
- Convert the configuration in XML or from database tables to a series of VCS One commands.

Please note the following limitations:

- Names of attributes may not exceed 32 characters.
- Values of attributes may not exceed 4096 bytes.
- Names of objects may not exceed 128 characters.
- Resource type names may not exceed 128 characters.

Note that the `haconf` command uses escape sequences for all special characters in XML files.

For the `-platform` option, supported values for `platform` are:

- aix
- aix/rs6000 (alias aix)
- esx
- hpux
- linux
- linux/x86 (alias linux)
- solaris
- solaris/x86
- solaris/sparc (alias solaris)
- windows
- windows/x86

For VMware ESX Server, use `linux` as the platform. Use the explicit platform name where no alias is defined. When `platform` appears in any displays, the full name and not the alias is shown.

OPTIONS

`-cleandb`

Clean the database before loading the configuration to the database (using the `-loaddb` option).

`-loaddb [-force] [`*`xml_dir`*`]`

Load the database by reading the XML configuration files in the *xml_dir* and writing the configuration to the database. The file main.xml must exist in the specified directory. The default directory for XML configuration files is:

UNIX and Linux: /etc/VRTSvcsone/conf/confxml

Windows Simulator: C:\Program Files\Veritas\Cluster Server One Simulator\conf\confxml

Windows: C:\Program Files\Veritas\Cluster Server One\conf\confxml

The command requires that you have write permission on *xml_dir* to run `haconf -loaddb`.

If you have not cleaned the database (using the `-cleandb` option), use the [`-force`] option to clean the database before loading it.

`-cftoxml` *cf_dir xml_dir* `-platform` *default_platform*

Convert specified VCS configuration files (.cf) in the directory *cf_dir* to XML format and place them in the directory *xml_dir*. Be advised that existing XML files of the same name in the specified directory are overwritten.

`-dbtoxml [-force]` *xml_dir*

Backup the current active configuration database to main.xml and types.xml files in the specified directory.

Caution: The command overwrites existing files using the same names.

If the configuration in the database is invalid, it is not backed up to xml. Use the `-force` option to bypass pre-backup verification of the database configuration. Doing so can be useful when fixing a corrupt configuration present in the database.

`-dbtocmd` *cmd_dir*

Converts the configuration in the database to a series of commands and places it to a file named config.cmd in the specified directory. Any existing config.cmd file in the specified directory is overwritten.

`-xmltocmd` *xml_dir*|*xml_file cmd_dir*

Converts an XML file, or directory that contains a configuration that is stored in XML files, to a series of commands. Dumps the commands to a file named config.cmd in the specified *cmd_dir*. In the case of the XML directory, the conversion includes the main.xml file and all included files. Be advised that any existing file named config.cmd in the *cmd_dir* directory is overwritten.

The command requires that you have write permission on *xml_dir* to run `haconf -xmltocmd`.

`-verify [`*xml_dir* `|-db]`

Verify the configuration files in the configuration directory (*xml_dir*) or in the database, using the `-db` option.

The command requires that you have write permission on *xml_dir* to run `haconf -verify` with the *xml_dir* option.

The default directory for XML configuration files is:

UNIX and Linux: /etc/VRTSvcsone/conf/confxml

Windows Simulator: C:\Program Files\Veritas\Cluster Server One Simulator\conf\confxml

Windows: C:\Program Files\Veritas\Cluster Server One\conf\confxml

`-version`

Display current version of `haconf` command.

`-dbstatus`

Displays the state of the database engine and the path of the configuration, if it is loaded. When the database is up, it is in the RUNNING state. Otherwise, the command reports the engine is not running or that it cannot connect to the database server.

`-help`

Display usage for `haconf` command. When you enter the command and an option without arguments, syntax for the specific option displays.

EXAMPLES

To load a configuration from the directory /tmp/myconfig to the database, use the `-force` option to clean it first:

```
haconf -loaddb /tmp/myconfig -force
```

Note: On Windows, a comparable directory to the /tmp directory that is used in these examples might be `C:\Windows\Temp`.

To convert the database configuration to a series of commands and place it in the file named config.cmd in the specified directory, enter:

```
haconf -dbtocmd /tmp/config_cmd
```

You can convert an XML file to a series of commands and place it in the config.cmd file in the specified directory. To do so, enter:

```
haconf -xmltocmd /tmp/ApacheTypes.xml /tmp/config_cmd
```

To convert the XML files in a configuration directory to a series of commands in the config.cmd file in the specified directory, enter:

```
haconf -xmltocmd /etc/VRTSvcsone/conf/confxml /tmp/config_cmd
```

Note: On Windows, the XML file path is `%VCSONE_HOME%\conf\confxml`, where `%VCSONE_HOME%` is the VCS OneVCS One installation directory.

To display the status of the database, enter a command similar to the following. The command output shows that the database is running with the loaded configuration:

```
haconf -dbstatus

VCSOne INFO V-97-1-17469 Database engine is RUNNING and loaded with

 configuration /etc/VRTSvcsone/conf/confxml

VCSOne INFO V-97-100-40 Database engine is RUNNING with complete

Rules and Jobs schema

VCSOne INFO V-97-102-1040 Database engine is RUNNING with complete

preferences schema
```

The command output shows that the database is not running:

```
haconf -dbstatus

VCSOne ERROR V-97-7-17 Unable to connect to database server.

VCSOne INFO V-97-1-17471 The database engine not running.

Start the database engine.
```

hacsg

`hacsg` – administers composite service groups in the VCS One cluster

SYNOPSIS

UNIX: /opt/VRTSvcsone/bin/hacsg

Windows: %VCSONE_HOME%\bin\hacsg

hacsg -add *csg_name* [*ouvaluepath*][-grp[-force] {*group(s)*|-ea *eaexpression*|-ou *ouexpression*|-ea *eaexpression* -ou *ouexpression*| -setname *setname*}] [-user *user@domain* -domaintype *domaintype*]
hacsg -delete *csg_name* [-user *user@domain* -domaintype *domaintype*]
hacsg -move [-updateroles] *csg_name(s)* -ou *ouvaluepath* [-user *user@domain* -domaintype *domaintype*]
hacsg -display [*csg_name(s)* | -ou *ouexpression*] [-attribute *attribute_name(s)*] [-user *user@domain* -domaintype *domaintype*]
hacsg -display [*csg_name(s)*] [-attribute *attribute_name(s)*] [-clus *cluster*] [-user *user@domain* -domaintype *domaintype*]
hacsg -value *csg_name_attribute*[-clus *cluster*][-user *user@domain* -domaintype *domaintype*]
hacsg -list *conditionals* [-user *user@domain* -domaintype *domaintype*]
hacsg -wait *csg_name attribute value*[time *seconds*][-clus *cluster*] [-user *user@domain* -domaintype *domaintype*]
hacsg -addgrp [-force] *csg_name* {*groups(s)* | -ea *eaexpression* | -ou *ouexpression* |-ea *eaexpression* -ou *ouexpression*|-setname *setname*} [-user *user@domain* -domaintype *domaintype*]
hacsg -deletegrp[-force]*csg_name* {*groups(s)*|-ea *eaexpression*|-ou *ouexpression*|-ea *eaexpression* -ou *ouexpression*|-setname *setname*} [-user *user@domain* -domaintype *domaintype*]
hacsg -groups *csg_name*[-user *user@domain* -domaintype *domaintype*]
hacsg -state[*csg_name(s)*|-ou *ouexpression*][-user *user@domain* -domaintype *domaintype*]
hacsg -state[*csg_name(s)*][-clus *cluster*][-user *user@domain* -domaintype *domaintype*]
hacsg -requestauth[-force]*csg_name* [-user *user@domain* -domaintype *domaintype*]
hacsg -online[-propagate][-force]*csg_name*[-user *user@domain* -domaintype *domaintype*]

```
hacsg -offline[-propagate]csg_name[-user user@domain -domaintype
domaintype]
hacsg -switch csg_name[-clus target_cluster][-user user@domain
-domaintype domaintype]
hacsg -infoattn csg_name[-user user@domain -domaintype domaintype]
hacsg -flush csg_name[-user user@domain -domaintype domaintype]
hacsg -modify modify_options
hacsg -help [-modify]
hacsg -version
```

AVAILABILITY

VRTSvcsonec

DESCRIPTION

The `hacsg` command administers composite service groups in the VCS One cluster. A composite service group is an object that groups together a set of service groups for disaster recovery operations.

`hacsg` Adds or deletes a composite service group, modifies the attributes of a composite service group, brings a composite service group online, or takes it offline. The `hacsg` command also performs the following functions:

- Switches a composite service group from one cluster to another from a local or a remote cluster
- Displays the service groups that are in a cluster service group
- Displays the attributes or attribute values for one or more composite service groups

A non-root user who has not run the `halogin` command can execute the `hacsg` command using the `-user` *user@domain* option. This option executes the command with the privileges of the specified user. When you issue the command, enter your fully qualified domain user name and supply a password when prompted. If necessary, the `-domaintype` option can specify the type of domain against which the user is to be authenticated. Supported domain types include:

- "unixpwd"
- "nt"
- "nis"
- "nisplus"
- "ldap"

- "pam"
- "vx" (Symantec Private Domain)

The default domain type is "vx". The domain type is case sensitive.

See NOTES for how to specify "-" and "%" characters in the command line.

OPTIONS

`-add` *`csg_name`*`[`*`ouvaluepath`*`][-grp[-force] {`*`group(s)`*`|-ea` *`eaexpression`*`| -ou` *`ouexpression`*`|-ea` *`eaexpression`* `-ou` *`ouexpression`* `| -setname` *`setname`*`} [-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Creates new composite service groups in the local VCS One cluster. Specify the composite service group names using *csg_name*. New composite service groups are attached at the specified *ouvaluepath*. A service group may be part of only one composite service group at a time.

Use the `-grp` option to specify the service groups to include in the composite service group. You can include an individual service group or multiple service groups. You can also include the service groups that you specify using one of the following:

- An OU expression
- An EA expression
- An OU expression and an EA expression
- A set

Use the `-force` option to force any service group(s) that can be added to the GroupList to be added.

`-delete` *`csg_name`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Deletes a composite service group that you specify using the *csg_name*. Deleting a composite service group does not delete the individual service groups within it.

`-move [-updateroles]` *`csg_name(s)`*`-ou` *`ouvaluepath`*`[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Moves a local composite service group to the OU Value node in the organization tree that you specify using `-ou` *ouvaluepath*.

A composite service group is attached to the organization tree at an OUValue node. The node where the composite service group is attached determines the user privileges that are associated with it.

Moving a composite service group can cause it to move outside of a user's home directory. In this situation, use the `-updateroles` option. This option deletes the composite service group from the user's role so that the user no longer has privileges on it. If you do not specify `-updateroles` in this situation, moving the composite service group is not allowed.

`-display [`*`csg_name(s)`*`|-ou `*`ouexpression`*`] [-attribute `*`attribute_name(s)`*`] [-user `*`user@domain`*` -domaintype `*`domaintype`*`]`

Displays the attribute names for the specified composite service group(s).

You can display the attribute names for an individual composite service group or multiple composite service groups. You can also display the attribute names for the composite service groups you specify using an OU expression.

`-display [`*`csg_name(s)`*` ][-attribute `*`attribute name(s)`*`][-clus `*`cluster`*`] [-user `*`user@domain`*` -domaintype `*`domaintype`*`]`

Displays the attribute names for the specified composite service group(s).

You may display the attribute names for an individual composite service group or multiple composite service groups.

You may also display the attribute names of a global composite service group that you configure on a VCS One cluster using the `-clus` *cluster* option. If you specify a local VCS One cluster, the command behavior is the same as if no cluster name is specified.

You see an error if the VCS One cluster that you specify using `-clus` *cluster* is not configured to communicate with the Policy Master in the local VCS Onecluster.

`-value `*`csg_name_attribute`*` [-clus `*`cluster`*`][-user `*`user@domain`*` -domaintype `*`domaintype`*`]`

Displays the attribute values for the specified composite service group in a local or a remote VCS One cluster.

You can display the attribute values of a global composite service group that you configured on a VCS One cluster using the `-clus` *cluster* option. If you specify a local VCS One cluster, the command behavior is the same as if no cluster name is specified.

You see an error if the VCS One cluster you specified using `-clus` *cluster* is not configured to communicate with the local VCS One cluster's Policy Master.

`-list `*`conditionals`*` [-user `*`user@domain`*` -domaintype `*`domaintype`*`]`

Lists all the composite service groups in the VCS One cluster. For global composite service groups, this command lists the names of the VCS One clusters in which the composite service groups are configured. The command lists global composite service groups for each VCS One cluster in the

ClusterList. It lists local composite service groups with `localclus` in the Cluster Name column.

The `-list` option accepts conditionals that are of the form `attr_name=attr_value`, where `attr_name` is a valid scalar-valued attribute for the CSG object.

`-wait` *`csg_name_attribute value`* [`-time` *`seconds`*][`-clus` *`cluster`*] [`-user` *`user@domain`* `-domaintype` *`domaintype`*]

The `-wait` option is for use in scripts. Use `-wait` with `hacsg` to wait until the attribute value has changed as specified, or until the duration that you specified in *seconds* has been reached. *seconds* is an integer specifying seconds. If you do not specify a value for *seconds*, `hacsg` waits indefinitely.

Use the `-wait` option only for changes to scalar attributes.

`-addgrp [-force]` *`csg_name`*`{`*`group(s)`*`|-ea` *`eaexpression`*`| -ou` *`ouexpression`*`|-ea` *`eaexpression`* `-ou` *`ouexpression`* `| -setname` *`setname`*`}` `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Adds service groups to a composite service group.

Use the `-force` option to force any service group(s) that can be added to the composite service group to be added. Groups that cannot be added are indicated in response messages.

You can add an individual service group or multiple service groups. You can also add the service groups that you specify using one of the following:

- A set
- An OU expression
- An EA expression
- An Ou expression and an EA expression

Use the -ou option to add service groups to the composite service group using an OU expression. Use the `-ea` option to add service groups using an EA expression.

By default, if any one group that you specify cannot be added, the operation fails and no groups are added.

`-deletegrp [-force]` *`csg_name`*`{`*`group(s)`*`|-ea` *`eaexpression`*`| -ou` *`ouexpression`*`|-ea` *`eaexpression`*`| -ou` *`ouexpression`* `| -setname` *`setname`*`}` `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Removes the service groups from the GroupList of a composite service group.

Use the -force option to force any group(s) that can be deleted from the composite service group to be deleted. Groups that cannot be deleted are indicated in response messages.

You can delete an individual service group or multiple service groups. You can also delete the service groups that you specify using:

- An OU expression
- An EA expression
- An OU expression and an EA expression

-ou Deletes from the composite service group the service groups that that the OU expression specifies. -ea Deletes from the composite service group the service groups that that the EA expression specifies.

By default, if any one of the specified service groups cannot be deleted, the operation fails and no groups are deleted.

If you delete the last service group from the composite service group, the composite service group remains, but is empty.

-groups *csg_name* [-user *user@domain* -domaintype *domaintype*]

Displays the names of the service groups in the composite service group.

-state [*csg_name(s)* | -ou *ouexpression*] [-user *user@domain* -domaintype *domaintype*]

Displays the state of the specified composite service group(s).

You can display the state for an individual composite service group or multiple composite service groups. You can also display the state for the composite service groups that an OU expression specifies.

-state [*csg_name(s)*] -clus *cluster* [-user *user@domain* -domaintype *domaintype*]

Displays the state of the specified composite service group(s). *cluster* Specifies the remote cluster. If you do not specify a composite service group, this option displays for all global composite service groups that are configured on the remote cluster.

-requestauth[-force]*csg_name* [-user *user@domain* -domaintype *domaintype*]

The specified composite service group requests to have authority over a local cluster. A remote cluster that has authority over a composite service group that is not online on the remote cluster relinquishes authority to the local cluster. If the composite service group is online on the remote cluster, the remote cluster does not relinquish authority. In that case, -requestauth fails.

`-force` Acquires authority for the composite service group in the local cluster if the remote cluster that has authority is not running or does not transition to a running state.

`-online[-propagate][-force]`*`csg_name`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Brings a composite service group online in the specified local cluster. A composite service group is online when all the service groups in it are online. This command option brings each service group in the composite service group online.

`-propagate` Brings online any offline child service groups that are:

- Outside of the composite service group
  Required to be online before you can bring the composite service group online

If you do not specify `-propagate`, the online operation on the composite service group fails, or partially succeeds if both of the following are true:

- Offline child service groups exist outside of the composite service group
- The offline child service groups that are outside of the composite service group must be online before the composite service group can be brought online

Use the `-force` option when:

- The cluster that has authority over the composite service group is disconnected or down
  You need to bring the composite service group online in the local cluster

`-offline [-propagate]`*`csg_name`*`[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Takes a composite service group offline in the specified local cluster. A composite service group is offline when all the service groups in it are offline. This command option takes each service group in the composite service group offline.

`-propagate` Takes offline any online firm and hard parent service groups for which both of the following are true:

- The parent service groups are outside of the composite service group
- The parent service groups have child service groups inside the composite service group

These parent service groups must be offline before the composite service group can go offline completely.

If you do not specify -propagate, the offline operation on the composite service group fails, or succeeds partially if both of the following are true:

- Firm and hard parent groups are online and outside of the composite service group
- The parent groups have child service groups in the composite service group

-switch *csg_name* -clus *target_cluster* [-user *user@domain* -domaintype *domaintype*]

Switches a composite service group from one cluster to another. You can use the -switch option on the cluster where the composite service group is online. You can also use the -switch option on the cluster where the composite service group goes online after the switch. On the target cluster, the state of the composite service group must be OFFLINE for the switch to succeed.

-infoattn *csg_name* [-user *user@domain* -domaintype *domaintype*]

Lists the reason that the ATTN flag is set in the CSGState attribute of a composite service group. Lists all of the groups in the composite service group that have caused the ATTN flag to be set and the reason. For example, the reason can be"Unable to Online" or "Group Fault". If a concurrency violation occurs for the composite service group, -infoattn lists only the composite service group name. It does not list a corresponding group name. The reason is " Concurrency Violation ".

-flush *csg_name* [-user *user@domain* -domaintype *domaintype*]

Flushes a composite service group. Flushing a composite service group clears all IntentOnline entries for any service groups in the composite service group.

-modify *modify_options*

The -modify option lets you modify a composite service group's attributes.

You may modify a scalar attribute's existing value.

You may not use modify to change values already defined for a vector, a keylist, or an association attribute. For vector, keylist, and association attributes, use the *modify_options*, which include -add, -delete, -update, or -delete -keys.

Refer to the following list of -modify commands. You may display the commands using hacsg -help -modify.

SCALAR

hacsg -modify *csg_name attribute value* [-user *user@domain* -domaintype *domaintype*]

VECTOR

Use the following command only when the attribute has no value:

```
hacsg -modify csg_name attribute value... [-user user@domain
-domaintype domaintype]
```

For vector attributes with defined values, only the following operations are allowed:

```
hacsg -modify csg_name attribute -add value... [-user
user@domain -domaintype domaintype]
```

```
hacsg -modify csg_name attribute -delete -keys [-user
user@domain -domaintype domaintype]
```

Note: You cannot delete an individual element of a VECTOR.

KEYLIST

Use the following command only when the attribute has no value:

```
hacsg -modify  csg_name attribute key... [-user user@domain
-domaintype domaintype]
```

For keylist attributes with defined values, only the following operations are allowed:

```
hacsg -modify csg_name attribute -add key... [-user
user@domain -domaintype domaintype]
```

```
hacsg -modify csg_name attribute -delete key... [-user
user@domain -domaintype domaintype]
```

hacsg –modify *csg_name attribute* –delete –keys [-user *user@domain* -domaintype *domaintype*]

ASSOCIATION

Use the following command only when the attribute has no value:

```
hacsg -modify csg_name attribute {key value}... [-user
user@domain -domaintype domaintype]
```

For association attributes with defined values, only the following operations are allowed:

```
hacsg -modify csg_name attribute -add {key value}... [-user
user@domain -domaintype domaintype]
```

```
hacsg -modify csg_name attribute -update {key value}... [-user
user@domain -domaintype domaintype]
```

```
hacsg -modify csg_name attribute -delete key... [-user
user@domain -domaintype domaintype]

hacsg -modify csg_name attribute -delete -keys [-user
user@domain -domaintype domaintype]
```

`-help [-modify]`

Displays the command usage for `hacsg`. Use `-help -modify` to display the command usage for `hacsg -modify`. When you enter `hacsg -help` and an option without arguments, the syntax for the specified option displays.

`-version`

Displays the command version.

EXAMPLES

To display the usage syntax for a specified command option, enter the command and an option without arguments. For example, to see the usage for `hacsg -addgrp`, enter:

```
# hacsg -addgrp
```

To bring a composite service group named csg_bigApp online in the local cluster, enter:

```
# hacsg -online csg_bigApp
```

To bring a composite service group named csg_bigApp and all its child service groups online in the local cluster, enter:

```
# hacsg -online -propagate csg_bigApp
```

To direct `hacsg` to wait until the CSGState attribute of a composite service group named csg_bigApp changes to the value ONLINE in the local cluster, enter:

```
# hacsg -wait csg_bigApp CSGState ONLINE
```

To switch a composite service group named csg_bigApp to a remote cluster named Cluster1, enter:

```
# hacsg -switch csg_bigApp -clus Cluster1
```

NOTES

When using the command to specify or modify an attribute value that begins with a dash ("–"), precede the value with a percent sign ("%"). For example, specify `-y` as `%-y`. Likewise, precede a value that starts with a percent sign with another percent sign.

SEE ALSO

`haclus`(1M), `hagrp`(1M)

hadb

`hadb` – manage the VCS One database

SYNOPSIS

```
UNIX: /opt/VRTSvcsone/bin/hadb

Windows: %VCSONE_HOME%\bin\hadb

hadb -backupxml backup_dir [-quiet | -verbose]
hadb -cleandb [-quiet | -verbose]
hadb -reloaddb backup_dir [-quiet | -verbose]
hadb -down [-quiet | -verbose]
hadb -dbpasswd [-quiet | -verbose]
hadb -initdb [-quiet | -verbose]
hadb -loaddb [-quiet | -verbose]
hadb -restart [-quiet | -verbose]
hadb -status
hadb -up [-quiet | -verbose]
hadb -version
hadb -help
For Windows only, an additional option is available:
hadb -uninstall
For the Simulator, the command usage is:
hadb -startsim [-cleandb] [-d xml_dir] [-extended [-no_operation]]
[-quiet | -verbose]
hadb -stopsim [-quiet | -verbose]
hadb -cleandb [-quiet | -verbose]
hadb -down [-quiet | -verbose]
hadb -initdb [-quiet | -verbose]
hadb -loaddb [-quiet | -verbose]
hadb -restart [-quiet | -verbose]
hadb -status
hadb -up [-quiet | -verbose]
hadb -help
```

AVAILABILITY

VRTSvcsonec, vcsonesim

DESCRIPTION

The `hadb` utility is for debugging and troubleshooting. When VCS One is up and running, there is no need to run this command.

The `hadb` utility provides the means to manage the VCS One configuration database. This database stores the VCS One configuration, which the Policy Master accesses when the VCS One cluster starts. The configuration is loaded initially from the XML files that are stored in the directory:

- UNIX and Linux: /etc/VRTSvcsone/conf/confxml
- Windows: C:\Program Files\Veritas\Cluster Server One\conf\confxml
- Windows Simulator: C:*installed_location*\VCS One\Simulator\conf\confxml
  Where *installed_location* is the location where you installed the Simulator. If you installed the Simulator in the default location, the sample configurations are located on your desktop under: \VCSOne\Simulator\conf\confxml

The utility also facilitates online backup of the database to XML files in a specified directory.

The `-quiet` and `-verbose` options specify what information is displayed as the command executes.

Do not use the `hadb` command when the Policy Master is running. The command can erase or bring down the database, and can cause the Policy Master to fail. Be sure to back up the configuration before running `hadb`.

OPTIONS

`-backupxml` *`backup_dir`* `[-quiet | -verbose]`

This command option is deprecated and replaced with `haconf -dbtoxml`. Symantec recommends, however, that you use `haadmin -backup` rather than `haconf -dbtoxml` to back up the configuration. See `haconf`(1M) and `haadmin`(1M) for more information.

`-cleandb [-quiet | -verbose]`

Clears the configuration in the database. Before loading a new configuration, use the `-cleandb` option with the Policy Master stopped.

This option removes configuration information from the database. Be sure to back up the configuration using `haadmin -backup` before using `hadb -cleandb`.

`-reloaddb` *`backup_dir`* `[-quiet | -verbose]`

Reload the database from the specified backup directory. Before loading a new configuration, stop the database daemon using `hadb -down`.

`-down [-quiet | -verbose]`

Stops the database daemon. Make sure that the Policy Master is not running when you issue this command.

`-dbpasswd [-quiet | -verbose]`

Changes the VCS One database password.

`-initdb [-quiet | -verbose]`

Initializes a database by creating new database files and transaction log files. Also resets the database password to the default value. To change the default password, use `hadb -dbpasswd`. Use `hadb -initdb` with caution because the existing database configuration is lost.

See additional option:

On Windows, `hadb -initdb` creates the VCS One Configuration Database service.

`-loaddb [-quiet | -verbose]`

Loads the database with the Policy Master configuration information in the XML files in:

UNIX: /etc/VRTSvcsone/conf/confxml

Windows: C:\Program Files\Veritas\Cluster Server One\conf\confxml

`-restart [-quiet | -verbose]`

Restarts the database. Make sure that the Policy Master is not running when you issue this command.

`-status`

Displays the current status of the VCS One database.

`-uninstall`

Available for the Windows Policy Master only, this option uninstalls the VCS One Configuration Database service. Make sure that the database service and Policy Master are not running when you issue this command.

`-up [-quiet | -verbose]`

Starts the database daemon if it is down. Make sure that the Policy Master is not running when you issue this command.

`-version`

Displays the current version of the `hadb` command.

`-help`

Display usage for `hadb` command.

The following command options apply for the Simulator:

`-startsim [-cleandb] [-d xml_dir] [-extended [-no_operation]] [-quiet | -verbose]`

In the Simulator, this command option loads the XML configuration that `-d` *xml_dir* specifies, and starts the Simulator. Specify the `-cleandb` option to clear the configuration in the database before loading a new configuration.

`-stopsim [-quiet | -verbose]`

In the Simulator, this command option stops the vcsoned process, the vcsonesim process, and the db process.

`-cleandb [-quiet | -verbose]`

In the Simulator, this command option clears the configuration in the database before it loads a new configuration. The `-cleandb` removes configuration information from the database.

`-down [-quiet | -verbose]`

In the Simulator, this command option stops the database. Stopping the database stops the configuration database along with its processes. Make sure that the Policy Master is not running when issuing this command.

`-initdb [-quiet | -verbose]`

In the Simulator, this command option initializes a database by creating new database files and transaction log files. Use `hadb -initdb` with caution because the existing database configuration is lost

`-loaddb [-quiet | -verbose]`

In the Simulator, this command option loads the database with the Policy Master configuration information in the XML files in:

UNIX: /etc/VRTSvcsone/conf/confxml

Windows: C:\Program Files\Veritas\Cluster Server One\conf\confxml

`-restart [-quiet | -verbose]`

In the Simulator, this command option restarts the database. Make sure that the Policy Master is not running when issuing this command.

`-status`

In the Simulator, this command option displays the current status of the VCS One database.

`-up [-quiet | -verbose]`

In the Simulator, this command option starts the database if it is down. Starting the database starts the database processes.

`-help`

In the Simulator, this command option displays usage for the `hadb` command.

EXAMPLES

Load the database from the default configuration directory.

```
hadb -loaddb
```

Reload the database from the specified backup directory.

```
hadb -reloaddb /usr/back/tmp -verbose
```

SEE ALSO

`haadmin`(1M), `haconf`(1M)

haea

`haea` – create and maintain extended attributes

SYNOPSIS

UNIX: /opt/VRTSvcsone/bin/haea

Windows: %VCSONE_HOME%\bin\haea

haea -add [-grp | -sys | -vobject | -pframe | -vframe] *ouvaluepath attribute* {{*values* [-default *value*]} | -freeform} [-desc *description*] [-user *user@domain* -domaintype *domaintype*]
haea -delete [-grp | -sys | -vobject | -pframe | -vframe] *attribute* [-user *user@domain* -domaintype *domaintype*]
haea -default [-grp | -sys | -vobject | -pframe | -vframe] [-propagate] [*ouvaluepath*] *attribute defaultvalue* [-user *user@domain* -domaintype *domaintype*]
haea -reset [-grp | -sys | -vobject | -pframe | -vframe] [-validvalues] *ouvaluepath attribute* [-user *user@domain* -domaintype *domaintype*]
haea -modify [-grp | -sys | -vobject | -pframe | -vframe] [*ouvaluepath*] *attribute* [-add [-propagate] | -delete | -update] *values* [-user *user@domain* -domaintype *domaintype*]
haea -updatedesc [-grp | -sys | -vobject | -pframe | -vframe] *attribute description* [-user *user@domain* -domaintype *domaintype*]
haea -display [-grp | -sys | -vobject | -pframe | -vframe | -all] [-definition] [-exclusive] *ouvaluepath* [-user *user@domain* -domaintype *domaintype*]
haea -value [-grp | -sys | -vobject | -pframe | -vframe] [-exclusive] *ouvaluepath attribute* [-user *user@domain* -domaintype *domaintype*]
haea -list [-grp | -sys | -vobject | -pframe | -vframe | -all] *ouvaluepath* [-user *user@domain* -domaintype *domaintype*]
haea -version
haea [-help]

AVAILABILITY

VRTSvcsonec

DESCRIPTION

The `haea` command is used to create and maintain extended attributes. You can define extended attributes at an OUValue node (specified by *ouvalue*) in the Organizational Tree. Defining extended attributes at an OUValue node makes them visible at all the OUValue nodes that are below the OUValue node where they are defined.

The properties for an extended attribute are:

`Form`: An extended attribute can be an enumerated form or freeform. An enumerated extended attribute has a set of valid values (called a validation set) that are defined as well as an optional default value. A freeform extended attribute, on the other hand, does not have a validation set or a default value.

`Type`: An extended attribute can have one of the following types: group, system, vobject, pframe, vframe, or common. An extended attribute of a given type is associated with an object of that type when it is attached to the *ouvalue* node. For example, an extended attribute of type system is associated with a system object when it is attached to the *ouvalue* node.

`Validation set`: A validation set defines a list of valid values that can be assigned to the extended attribute's value for a group, system, vobject, pframe, or vframe object. A validation set that is at a lower level *ouvalue* node is always both of the following:

- A subset of the validation set that is of the extended attribute where it is first defined
- A subset of its parent *ouvalue* node where it is overridden

`Default value`: A default value can be specified for an extended attribute at an *ouvalue* node. The default value is automatically assigned to a group, system, vobject, pframe, or vframe object when it is associated with this extended attribute for the first time.

An OU expression and an organization unit value cannot contain spaces.

An extended attribute value cannot contain a comma.

An EA expression must be enclosed in double quotes if it contains spaces.

In addition, an extended attribute value or validation set cannot contain a single quote (') character. Single quotes are used to enclose extended attribute values of more than one word in an EA expression. For example:

```
hagrp -display -ea "ea1= 'new value' and ea2='new value2'"
```

OPTIONS

`-add [-grp | -sys | -vobject | -pframe | -vframe] *ouvaluepath attribute* {{*values* [-default *value*]} | -freeform} [-desc *description*] [-user *user@domain* -domaintype *domaintype*]`

Adds a group-type (`-grp` option), a system-type (`-sys` option), a vobject-type (`-vobject` option), a pframe-type (`-pframe` option), a vframe-type (`-vframe` option) or a common-type (default) extended attribute at a specified *ouvaluepath* node. *ouvaluepath* is the location of the OUValue to which the attribute is attached as denoted by an Organization Tree path.

Use the `-freeform` option if the extended attribute will have freeform values.

If you do not use the `-freeform` option, specify multiple space-separated values. Freeform extended attributes do not use a validation set or a default value. Enumerated attributes do.

Use the `-desc` option to specify a description.

`-delete [-grp | -sys | -vobject | -pframe | -vframe] *attribute* [-user *user@domain* -domaintype *domaintype*]`

Deletes a group, system, vobject, pframe, vframe, or common type extended attribute. This operation deletes the specified extended attribute for all the object instances and *ouvalue* nodes wherever they are used. *attribute* is the name of the attribute to be deleted.

`-default [-grp | -sys | -vobject | -pframe | -vframe] [-propagate] [*ouvaluepath*] *attribute defaultvalue* [-user *user@domain* -domaintype *domaintype*]`

Specifies the default value for the extended attribute. The *ouvaluepath* sets the default value for the extended attribute at the specified *ouvaluepath*. A default value has to be part of the validation set. If you set an extended attribute default to an *ouvalue* node other than the one where the extended attribute is defined, you override the default value. Changes to an extended attribute default value apply to all the child nodes in the Organization Tree that do not override the default value.

`-reset [-grp | -sys | -vobject | -pframe | -vframe] [-validvalues] *ouvaluepath attribute* [-user *user@domain* -domaintype *domaintype*]`

Resets the default value of the extended attribute to the default value that is defined in the parent *ouvalue* node's extended attribute. The `-validvalues` option can also reset the validation set that is specified in the parent *ouvalue* node's extended attribute.

`-modify [-grp | -sys | -vobject | -pframe | -vframe] [`*`ouvaluepath`*`] attribute -add [-propagate] | -delete | -update] values [-user user@domain -domaintype domaintype]`

Modifies the validation set that is part of the extended attribute at the specified *ouvaluepath*. Setting an extended attribute's validation set to an *ouvalue* node other than the one where the extended attribute is defined, overrides the validation set. The change is applied to all the extended attributes down the Organization Tree until the validation set is not overridden. If `-propagate` is used with `-add`, the new value is added at all the extended attributes below the specified node. The `-delete` option can be used to delete the values at the specified node and below. The `-update` option can be used to update the values at the specified node and below. By default, (that is, if the `-add` or `-delete` options are not specified), the validation set is updated at the specified node and below. *values* is a space-delimited list of the values to be modified.

`-display [-grp | -sys | -vobject | -pframe | -vframe | -all] [-definition] [-exclusive] ouvaluepath [-user user@domain -domaintype domaintype]`

Displays the extended attributes information for the specified *ouvaluepath* and below. By default, common extended attributes are displayed. If the `-all` option is specified, all types of extended attributes are displayed. Use the `-definition` option to display the definitions of extended attributes. The `-exclusive` option can be used to display information only at the specified *ouvaluepath*.

`-value [-grp | -sys | -vobject | -pframe | -vframe] [-exclusive] ouvaluepath attribute [-user user@domain -domaintype domaintype]`

Displays the default value for an extended attribute at the specified *ouvaluepath*. Use the `-exclusive` option to display the default value of the extended attribute solely for the specified node. If you do not include the `-exclusive` option, the default value of all the extended attributes at and below the specified node are displayed.

`-list [-grp | -sys | -vobject | -pframe | -vframe | -all] ouvaluepath [-user user@domain -domaintype domaintype]`

Lists the extended attributes and the *ouvaluepath* where they are defined. The `-list` option also displays the description of the extended attribute.

`-version`

Displays version of the command.

`[-help]`

Displays usage for the `haea` command.

EXAMPLES

To create a new group-type extended attribute, enter, for example:

```
# haea -add -grp / location NY Mumbai SFO -default NY
```

To create a new system-type extended attribute, enter, for example:

```
# haea -add -sys /lob=dcmg MACAddress -freeform
```

To create a new pframe-type extended attribute, enter, for example:

```
# haea -add -pframe / Location Lab1 Lab2 Lab3 -default Lab1
```

Adds an extended attribute for pframe objects at the organization unit "/" with the options "Lab1", "Lab2", and "Lab3" with a default value of "Lab 1".

To display all extended attributes, enter:

```
# haea -display -all
```

```
Extended Attribute for OUValuePath /
```

```
-----------------------------------
```

#Attribute	Type	Flags	DefaultValue	ValidValues
location	Group	Enumerated	NY	NY Mumbai SFO

```
Extended Attribute for OUValuePath /lob=dcmg
```

```
-------------------------------------------
```

#Attribute	Type	Flags	Default value	ValidValues
location	Group	Enumerated	NY	NY Mumbai SFO
MACAddress	System	FreeForm		

```
Extended Attribute for OUValuePath /lob=dcmg/dept=vcs
```

```
-----------------------------------------------------
```

#Attribute	Type	Flags	DefaultValue	ValidValues

```
location        Group       Enumerated    NY          NY Mumbai SFO
MACAddress      System      FreeForm

Extended Attribute for OUValuePath /lob=dcmg/dept=vcsone
--------------------------------------------------------
#Attribute      Type        Flags         DefaultValue  ValidValues
location        Group       Enumerated    NY          NY Mumbai SFO
MACAddress      System      FreeForm

Extended Attribute for OUValuePath /lob=consumer
------------------------------------------------
#Attribute      Type        Flags         DefaultValue  ValidValues
location        Group       Enumerated    NY          NY Mumbai SFO
```

To list all extended attributes, extended attribute types, and OUValuePaths, enter:

```
# haea -list -all /
#Attribute      Type        OUValuePath     Description
location        Group       /
MACAddress      System      /lob=dcmg
```

To create a default group type extended attribute, enter, for example:

```
# haea -default -grp /lob=dcmg/dept=vcs Location SFO
```

To modify a group type extended attribute, enter, for example:

```
# haea -modify -grp /lob=dcmg/dept=vcsone Location Mumbai SFO
```

To reset a group type extended attribute, enter, for example:

```
# haea -reset -grp /lob=dcmg/dept=vcs Location
```

To reset the validation set for a group type extended attribute as specified for the parent OUValue nodes extended attribute, enter, for example:

```
# haea -reset -grp -validvalues /lob=dcmg/dept=vcsone Location
```

To delete an extended attribute, enter, for example:

```
# haea -delete -sys MACAddress
```

NOTES

When using the command to specify or modify an attribute value that begins with a dash ("-"), precede the value with a percent sign ("%"). For example, specify `-y` as `%-y`. Likewise, precede a value that starts with a percent sign with another percent sign.

SEE ALSO

`haou`(1M), `haset`(1M)

haencrypt

`haencrypt` – generate encrypted passwords for use in VCS One configurations

SYNOPSIS

```
UNIX: /opt/VRTSvcsone/bin/haencrypt

Windows: %VCSONE_HOME%\bin\haencrypt

haencrypt -agent [password|-file file [-delete]]
haencrypt [-help]
```

AVAILABILITY

VRTSvcsonec

DESCRIPTION

The `haencrypt` command can be used to generate encrypted passwords. The command prompts you to enter a password and returns an encrypted password. You can use encrypted passwords when you configure the resources that require password information.

OPTIONS

`-agent [`*`password`*`|file` *`file`* `[-delete]]`

The `-agent` option without additional options, prompts you for your VCS One password and returns the password in an encrypted form. You can manually enter the encrypted password in the main.xml file as a value for a given resource's password attribute. You can also enter it dynamically as an attribute value from the command line when modifying resource attributes. You do not need to encrypt the password when you enter it through the GUI.

The `-file` *file* option reads the password that is encrypted from the specified file. Specify the `-delete` option with `-file` *file* if you want `haencrypt` to delete the file after reading the password from it.

`[-help]`

Displays usage for the `haencrypt` command.

EXAMPLES

You can generate an encrypted password that you can enter manually in main.xml as a value for a given resource's password attribute. You can also generate an encrypted password dynamically as an attribute value from the command line when you modify resource attributes. To generate an encrypted password, enter the following:

```
# haencrypt -agent

Enter New Password:

Enter Again:

hvpVqvR
```

The password that you enter is not displayed on the console.

SEE ALSO

`haattr`(1M)

hapframe

hapframe – add, modify, or delete the physical systems that are dedicated for virtualization (pframes), and display or list information about the physical systems that are dedicated for virtualization

SYNOPSIS

UNIX: /opt/VRTSvcsone/bin/hapframe

Windows: %VCSONE_HOME%\bin\hapframe

hapframe -add *pframe* -vtype *vtypename* [ouvaluepath] [-user *user@domain* -domaintype *domaintype*]
hapframe -delete *pframe* [-user *user@domain* -domaintype *domaintype*]
hapframe -move [-updateroles] [-refreshvars] *pframe(s)* -ou *ouvaluepath* [-user *user@domain* -domaintype *domaintype*]
hapframe -freeze [-evacuate] {[*pframe(s)* | -ou *ouexpression* [-info] | -ea *eaexpression* [-info] | -ou *ouexpression* -ea *eaexpression* [-info] | -setname *setname* [-info]} [-user *username@domain* -domaintype *domaintype*]
hapframe -unfreeze {[*pframe(s)* | -ou *ouexpression* [-info] | -ea *eaexpression* [-info] | -ou *ouexpression* -ea *eaexpression* [-info] | -setname *setname* [-info]} [-user *username@domain* -domaintype *domaintype*]
hapframe -display [*pframe(s)* | -ou *ouexpression* | -ea *eaexpression* | -ou *ouexpression* -ea *eaexpression* | -setname *setname*] [-attribute *attribute(s)*] [-user *username@domain* -domaintype *domaintype*]
hapframe -displayea [*pframe(s)*] [-attribute *attribute(s)*] [-user *user@domain* -domaintype *domaintype*]
hapframe -list [-vtype *vtype*] [*conditional(s)*] [-user *username@domain* -domaintype *domaintype*]
hapframe -clientversion [*pframe(s)*] [-user *user@domain* -domaintype *domaintype*]
hapframe -state [*pframe(s)* | -ou *ouexpression* | -ea *eaexpression* | -ou *ouexpression* -ea *eaexpression* | -setname *setname*] [-user *username@domain* -domaintype *domaintype*]
hapframe -associate *pframe* [-user *username@domain* -domaintype *domaintype*]
hapframe -disassociate *pframe* [-user *username@domain* -domaintype *domaintype*]

```
hapframe -value pframe attribute [-user username@domain -domaintype
domaintype]
hapframe -infovars system attribute [key] [-user user@domain
-domaintype domaintype]
hapframe -nodeid [nodeid] [-user username@domain -domaintype
domaintype]
hapframe -readconfig pframe [-user username@domain -domaintype
domaintype]
hapframe -fault pframe [-user username@domain -domaintype domaintype]
hapframe -wait pframe attribute value [-time seconds] [-user
username@domain -domaintype domaintype]
hapframe -modify modify_options
hapframe [-help [-modify | -list]]
hapframe -version
```

AVAILABILITY

VRTSvcsonec

DESCRIPTION

The `hapframe` command allows administrators to manage information about each physical system that hosts virtual machines (pframes). These pframes are the nodes that run the VCS One client daemon.

OPTIONS

`-add pframe -vtype vtypename [ouvaluepath] [-user username@domain -domaintype domaintype]`

Add a pframe *pframe* to the VCS One cluster. Do not use the word pframe to name a physical system or a pframe, VCS One reserves its use.

Specify the *vtype*. Use the `-vtype` *vtypename* option to specify the virtualization technology for the pframe. The accepted value for *vytpename* is esxserver or ldomserver.

You may optionally specify *ouvaluepath*. If you do not specify an OUValuePath (*ouvaluepath*), the pframe is added to the root (/) of the Organization Tree.

The pframe that this object represents does not need to exist or be a part of the VCS One cluster when you issue the command. The pframe that *pframe* specifies does not need to correspond to the host name of the actual pframe. However, it is recommended that you match the pframe with the hostname.

If security is enabled, it is almost essential that *pframe* matches the fully qualified host name of the pframe in question.

`-delete` *`pframe`* `[-user` *`username@domain`* `-domaintype` *`domaintype`*`]`

Delete a pframe from the configuration. The pframe must not be running the VCS One client daemon. Use `hastop -sys` to stop the VCS One client daemon on the pframe.

`-move [-updateroles] [-refreshvars]` *`pframe(s)`* `-ou` *`ouvaluepath`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Move a specified pframe or pframes in the VCS One configuration. Moving a pframe can cause the system to move outside of a user's home node. In this situation, use the `-updateroles` option. This option deletes the pframe from the user 's role so that the user no longer has privileges on that pframe. If you do not specify `-updateroles`, the pframe move is not allowed. If you attempt to move a pframe and if the current value of any of its extended attributes (that are used as resource variables) changes at the new location, the move is rejected. To override this behavior and move the pframe, use`-refreshvars`. Doing so will modify the value of the resource attributes that use the variable.

`-freeze [-evacuate] [`*`pframe(s)`* `| -ou` *`ouexpression`* `| -ea` *`eaexpression`* `| -ou` *`ouexpression`* `-ea` *`eaexpression`* `| -setname` *`setname`*`] [-user` *`username@domain`* `-domaintype` *`domaintype`*`] [-info]`

Freeze one or more pframes. The command freezes the pframes you specify using:

- An OU expression (*ouexpression*)
- An EA expression (*eaexpression*)
  An OU expression (*ouexpression*) and an EA expression (*eaexpression*)
  A set (*setname*)

The vframes that are configured on the frozen pframe cannot come online. They cannot come online manually, by failover, or by switching until the pframe is thawed. Thaw the pframe using the `-unfreeze` option. -evacuate Specifies that all vframes are switched before the pframe is frozen; if no other pframe is available for a vframe, it is taken offline. The vframes running on other pframes do not fail over to a frozen pframe.

`-info` Displays the objects that the command acts upon if executed. When `-info` is specified, the command is not executed; only information is displayed.

`-unfreeze [pframe(s) | -ou ouexpression | -ea eaexpression | -ou ouexpression -ea eaexpression | -setname setname] [-user username@domain -domaintype domaintype] [-info]`

Unfreeze one or more pframes that you specify using:

- An OU expression (*ouexpression*)
- An EA expression (*eaexpression*)
- An OU expression (*ouexpression*) and an EA expression (*eaexpression*)
- A set (*setname*)

`-display [pframe(s) | -ou ouexpression | -ea eaexpression | -ou ouexpression -ea eaexpression | -setname setname] [-attribute attribute(s)] [-user username@domain -domaintype domaintype]`

Display the attribute names and their values for a specified pframe or pframes that you specify using:

- An OU expression (*ouexpression*)
- An EA expression (*eaexpression*)
- An OU expression (*ouexpression*) and an EA expression (*eaexpression*)
- A set (*setname*)

If no pframe is specified, the attributes and values for all pframes are displayed.

An OU expression cannot contain spaces.

An EA expression must be enclosed in double quotes if it contains spaces.

An extended attribute value cannot contain a comma.

In addition, an extended attribute value or validation set cannot contain a single quote (') character. The single quote character serves as a delimiter for the value in an EA expression. However, you can use single quotes to enclose an extended attribute value with multiple words in an EA expression. For example:

```
hapframe -display -ea "ea1= 'new value' and ea2= 'new value2'"
```

`-displayea [pframe(s)] [-attribute attribute(s)] [-user user@domain -domaintype domaintype]`

Display the extended attributes and their values for a specified pframe or pframe(s). If no pframe is specified, the extended attributes and values for all pframes are displayed.

`-list [-vtype vtype] [conditional(s)] [-user user@domain -domaintype domaintype]`

Displays a list of pframes whose values match given conditional statement(s). Conditional statements can take three forms: Attribute=Value, Attribute!=Value, Attribute=~Value. Multiple conditional statements imply AND logic. The command lists all pframes in the VCS One cluster when no conditional statement is used.

For example, `hapframe -list PlatformName=esx` lists all the pframes where the PlatformName attribute value contains `esx`.

Use the `-vtype` option to display a list of pframes of a given vtype.

For example, `hapframe -list -vtype esxserver` lists all the pframes that have the vtype of `esxserver`.

`-clientversion [pframe(s)] [-user user@domain -domaintype domaintype]`

Displays the version of the client daemon that is installed on the pframe.

`-state [pframe(s) | -ou ouexpression | -ea eaexpression | -ou ouexpression -ea eaexpression | -setname setname] [-user username@domain -domaintype domaintype]`

Display the current state of the specified pframes. Specify pframes using

- An OU expression (*ouexpression*)
- An EA expression (*eaexpression*)
- An OU expression (*ouexpression*) and an EA expression (*eaexpression*)
- A set (*setname*)

The command displays the states of all pframes if you do not specify any pframes.

`-associate pframe objectname [-user username@domain -domaintype domaintype]`

`-associate` Builds associations between pframes and management servers (like VirtualCenter). The associations help with visualization of the VCS One cluster as well as enabling certain commands.

`-disassociate pframe objectname [-user username@domain -domaintype domaintype]`

`-disassociate` Severs associations between pframes and management servers (like VirtualCenter).

`-value pframe attribute`

The `-value` option provides the value of a single pframe attribute. For example, `hapframe -value esxb SysState` displays the value of the `SysState`

attribute for pframe `esxb`. `-value` Shows the value of one specific attribute rather than a table of the many attribute values that the `-display` option shows.

See EXAMPLES.

`-infovars` *`system attribute`* [*`key`*] [`-user` *`user@domain`* `-domaintype` *`domaintype`*]

Displays the resource attributes that use the specified attribute as a variable.

See EXAMPLES.

`-nodeid` [*`nodeid`*]

Return the current node name and nodeid values for the specified pframe. Values for the current pframe are returned if *nodeid* is not provided.

`-fault` *`pframe`* [`-user` *`username@domain`* `-domaintype` *`domaintype`*]

Can be used to force the client to a FAULTED state if it is in the DDNA state. The `-fault` option cannot be used if the client pframe is in the RUNNING state.

`-readconfig` *`pframe`* [`-user` *`username@domain`* `-domaintype` *`domaintype`*]

`-readconfig` Resets the configuration of a pframe without restarting the VCS One client. It reads the pframe's configuration file, vcsone.conf, for any updates. You can only change the SystemIPAddrs property in the vcsone.conf file. For example, if a pframe gets a new IP address, you can edit the SystemIPAddrs property in th pframe's configuration file and issue this command. Refer to the *Veritas Cluster Server One User's Guide* for the syntax of the configuration file.

`-wait` *`pframe attribute value`* [`-time` *`seconds`*]

The `-wait` option is for use in scripts to direct the `hapframe` command to wait until one of the following happens:

- The value of the attribute is changed as specified
- The specified number of *seconds* has elapsed

seconds is an integer specifying seconds. If *seconds* is not specified, `hapframe` waits indefinitely.

The `-wait` option can only be used with changes to scalar attributes.

See EXAMPLES.

`-modify` *`modify_options`*

The `-modify` option lets you modify a pframe's attributes. Some attributes are internal to VCS One and cannot be modified. You can modify any attribute that can be configured in main.xml.

You may modify a scalar attribute's existing value.

You may not use `-modify` to change values already defined for a vector, a keylist, or an association attribute. For vector, keylist, and association attributes, use the *modify_options*, which include `-add`, `-delete`, `-update`, or `-delete -keys`.

Refer to the following list of permissible `-modify` commands. You may display the commands by using `hapframe -help -modify`.

`SCALAR`

`hapframe -modify [-refreshvars] pframe attribute value [-user username@domain -domaintype domaintype]`

If you attempt to modify an extended attribute value that is a variable, an error message displays and the value is not modified. To override this behavior and modify an extended attribute value that is a variable, use the `-refreshvars` option. Doing so modifies the value of the resource attributes that use the variable.

`VECTOR`

Use the following command only when the attribute has no value:

`hapframe -modify pframe attribute value ... [-user username@domain -domaintype domaintype]`

For the vector attributes that have values defined, only the following operations are allowed.

`hapframe -modify pframe attribute -add value ... [-user username@domain -domaintype domaintype]`

`hapframe -modify pframe attribute -delete -keys [-user username@domain -domaintype domaintype]`

Note: You cannot delete an individual element of a VECTOR.

`KEYLIST`

Use the following command only when the attribute has no value:

`hapframe -modify pframe attribute key ... [-user username@domain -domaintype domaintype]`

For the keylist attributes that have values defined, only the following operations are allowed.

`hapframe -modify pframe attribute -add key ... [-user username@domain -domaintype domaintype]`

```
hapframe -modify pframe attribute -delete key ... [-user
username@domain -domaintype domaintype]

hapframe -modify pframe attribute -delete -keys [-user
username@domain -domaintype domaintype]
```

ASSOCIATION

Use the following command only when the attribute has no value:

```
hapframe -modify pframe attribute {key value} ... [-user
username@domain -domaintype domaintype]
```

For the association attributes that have values defined, only the following operations are allowed.

```
hapframe -modify pframe attribute -add {key value} ... [-user
username@domain -domaintype domaintype]

hapframe -modify pframe attribute -update {key value} ...
[-user username@domain -domaintype domaintype]

hapframe -modify pframe attribute -delete key ... [-user
username@domain -domaintype domaintype]

hapframe -modify pframe attribute -delete -keys [-user
username@domain -domaintype domaintype]
```

[-help [-modify | -list]]

The -help option displays the command usage for hapframe. The -modify option displays the usage for the -modify option. The -list option displays the usage for the -list option. When you enter the command and an option without arguments, syntax for the specific option displays.

-version

Display the version of hapframe.

EXAMPLES

Example 1. To display the usage syntax for a specific command option, enter the command and an option without arguments. For example, enter:

```
# hapframe -value
```

Example 2. From a script, to use the -wait option to direct the hapframe command to block until pframe P1 goes into the RUNNING state, enter:

```
# hapframe -wait P1 SysState RUNNING
```

NOTES

If a pframe name is not specified, information regarding all pframes is displayed.

If an attribute name is not specified, information regarding all pframe attributes is displayed.

When using the command to specify or modify an attribute value that begins with a dash ("-"), precede the value with a percent sign ("%"). For example, specify `-y` as `%-y`. Likewise, precede a value that starts with a percent sign with another percent sign.

SEE ALSO

`halogin`(1M),`haconf`(1M),`haclus`(1M),`havframe`(1M)

havframe

havframe – havframe - add, modify, or delete a virtual machine, and display or list information about virtual machines

SYNOPSIS

UNIX: /opt/VRTSvcsone/bin/havframe

Windows: %VCSONE_HOME%\bin\havframe

havframe -add *vframe* vtype *vtypename* [*ouvaluepath*] [-user *user@domain* -domaintype *domaintype*]
havframe -delete [-force] *vframe* [-user *user@domain* -domaintype *domaintype*]
havframe -move [-updateroles] [-refreshvars] *vframes* -ou *ouvaluepath* [-user *user@domain* -domaintype *domaintype*]
havframe -compatible [-propagate] *vframe1 vframe2* [-user *user@domain* -domaintype *domaintype*]
havframe -compatible [-propagate] -setname *setname* -withsetname *setname* [-user *user@domain* -domaintype *domaintype*] [-info]
havframe -compatible [-propagate] {-ou *ouexpression* | -ea *eaexpression* | -ou *ouexpression* -ea *eaexpression*} {-withou *ouexpression* | -withea *eaexpression* | -withou *ouexpression* -withea *eaexpression*} [-user *user@domain* -domaintype *domaintype*] [-info]
havframe -compatible [-propagate] *vframe* ALLVFRAMES [-user *user@domain* -domaintype *domaintype*]
havframe -incompatible [-propagate] *vframe1 vframe2* [-user *user@domain* -domaintype *domaintype*]
havframe -incompatible [-propagate] *vframe* ALLVFRAMES [-user *user@domain* -domaintype *domaintype*]
havframe -incompatible [-propagate] -setname *setname* -withsetname *setname* [-user *user@domain* -domaintype *domaintype*] [-info]
havframe -incompatible [-propagate] {-ou *ouexpression* | -ea *eaexpression* | -ou *ouexpression* -ea *eaexpression*} {-withou *ouexpression* | -withea *eaexpression* | -withou *ouexpression* -withea *eaexpression*} [-user *user@domain* -domaintype *domaintype*] [-info]
havframe -link *parentvframe childvframe* | *childgroup relationship* [-user *user@domain* -domaintype *domaintype*]
havframe -link *parentgroup childvframe relationship* [-user *user@domain* -domaintype *domaintype*]

```
havframe -unlink parentvframe childvframe | childgroup [-user
user@domain -domaintype domaintype]
havframe -unlink parentgroup childvframe [-user user@domain
-domaintype domaintype]
havframe -dep [vframe(s)] [-user user@domain -domaintype domaintype]
havframe -linksys [-force] vframe system [-user user@domain
-domaintype domaintype]
havframe -unlinksys vframe system [-user user@domain -domaintype
domaintype]
havframe -associate vframe objectname [-user user@domain -domaintype
domaintype]
havframe -disassociate vframe objectname [-user user@domain
-domaintype domaintype]
havframe -clear {vframe | -setname setname | -ou ouexpression | -ea
eaexpression | -ou ouexpression -ea eaexpression} [-pframe pframe]
[-user user@domain -domaintype domaintype]
havframe -clearadminwait [-fault] vframe -pframe pframe [-user
user@domain -domaintype domaintype]
havframe -flush [-action] vframe -pframe pframe [-user user@domain
-domaintype domaintype]
havframe -flush [-intent] vframe [-user user@domain -domaintype
domaintype]
havframe -online [{-ejectlowpri [-ignorestandby]} | -ignorestandby
| -propagate] vframe -pframe pframe [-user user@domain -domaintype
domaintype]
havframe -online [-ejectlowpri] [-nointent] vframe(s) -any [-user
user@domain -domaintype domaintype]
havframe -online [-ejectlowpri] [-nointent] {-setname setname | -ou
ouexpression | -ea eaexpression | -ou ouexpression -ea eaexpression}
-any [-user user@domain -domaintype domaintype] [-info]
havframe -offline [-propagate | -stopapps] vframe [-pframe pframe]
[-user user@domain -domaintype domaintype]
-offline [-propagate] vframe -everywhere [-user user@domain-domaintype
domaintype]
havframe -offline {-setname setname | -ou ouexpression | -ea
eaexpression | -ou ouexpression -ea eaexpression} -everywhere [-info]
[-user user@domain -domaintype domaintype]
havframe -offline -force vframe -pframe pframe [-user user@domain
-domaintype domaintype]
```

havframe -switch [-ejectlowpri | -propagate] [-ignorestandby] *vframe* -to *pframe* [-user *user@domain* -domaintype *domaintype*]
havframe -switch [-ejectlowpri] *vframe* -any [-user *user@domain* -domaintype *domaintype*]
havframe -migrate [-ejectlowpri | -propagate] [-ignorestandby] *vframe* -to *pframe* [-user *user@domain* -domaintype *domaintype*]
havframe -freeze [-propagate] *vframe* [-user *user@domain* -domaintype *domaintype*]
havframe -unfreeze [-propagate] *vframe* [-user *user@domain* -domaintype *domaintype*]
havframe -enable *vframe(s)* [-pframe *pframe*] [-user *user@domain* -domaintype *domaintype*]
havframe -enable {-setname *setname* | -ou *ouexpression* | -ea *eaexpression* | -ou *ouexpression* -ea *eaexpression*} [-pframe *pframe*] [-user *user@domain* -domaintype *domaintype*]
havframe -enable -all [-pframe *pframe*] [-user *user@domain* -domaintype *domaintype*]
havframe -disable *vframe(s)* [-pframe *pframe*] [-user*user@domain* -domaintype *domaintype*]
havframe -disable {-setname *setname* | -ou *ouexpression* | -ea *eaexpression* | -ou *ouexpression* -ea *eaexpression*} [-pframe *pframe*] [-user *user@domain* -domaintype *domaintype*]
havframe -disable -all [-pframe *pframe*] [-user *user@domain* -domaintype *domaintype*]
havframe -enableresources *vframe* [-user *user@domain* -domaintype *domaintype*]
havframe -disableresources *vframe* [-user *user@domain* -domaintype *domaintype*]
havframe -resources *vframe* [-user *user@domain* -domaintype *domaintype*]
havframe -changeload [-ejectlowpri | -tryswitch] *vframe* {*key value*} ... [-user *user@domain* -domaintype *domaintype*]
havframe -display [*vframe(s)* | -ou *ouexpression* | -ea *eaexpression* | -ou *ouexpression* -ea *eaexpression* | -setname *setname*] [-attribute *attribute(s)*] [-pframe *pframe(s)*] [-user *user@domain* -domaintype *domaintype*]
havframe -displayea [*vframe(s)*] [-attribute *attribute(s)*] [-user *user@domain* -domaintype *domaintype*]
havframe -list [-vtype *vtype*] [*conditional(s)*] [-user *user@domain* -domaintype *domaintype*]

```
havframe -state [vframe(s) | -setname setname | -ou ouexpression |
-ea eaexpression | -ou ouexpression -ea eaexpression] [-pframe
pframe(s)] [-user user@domain -domaintype domaintype]
havframe -value vframe attribute [-pframe pframe] [-user user@domain
-domaintype domaintype]
havframe -infovars vframe attribute [key] [-user user@domain
-domaintype domaintype]
havframe -wait vframe attribute value [-pframe {pframe | -any}] [-time
seconds] [-user user@domain -domaintype domaintype]
havframe -addpframe [-propagate] vframe pframe(s) [-user user@domain
-domaintype domaintype]
havframe -modify modify_options
havframe [-help [-modify | -link | -list]]
havframe -version
```

AVAILABILITY

VRTSvcsonec

DESCRIPTION

A vframe is a virtual machine that VCS One makes highly available. Use the `havframe` command to manage virtual machines and to view information about them.

An OU expression cannot contain spaces.

An EA expression must be enclosed in double quotes if it contains spaces.

An extended attribute value cannot contain a comma.

In addition, an extended attribute value or validation set cannot contain a single quote (') character. The single quote character serves as a delimiter for the value in an EA expression. However, single quotes can be used to enclose an extended attribute value that has multiple words in an EA expression. For example:

```
havframe -display -ea "ea1= 'new value' and ea2= 'new value2'"
```

An organization unit value cannot contain spaces.

For the `-vtype` option, supported values for *vtypename* are:

```
esxvm
```

A non-root user who has not run the `halogin` command can execute the `havframe` command using the `-user` *user@domain* option. This option executes the command

with the privileges of the specified user. When you issue the command, the user must enter the fully qualified domain user name and supply a password when prompted. If necessary, the `-domaintype` option can specify the type of domain against which the user is to be authenticated. Supported domain types include:

- "unixpwd"
- "nt"
- "nis"
- "nisplus"
- "ldap"
- "pam"
- "vx" (Symantec Private Domain)

The default domain type is "vx". The domain type is case sensitive.

See NOTES for how to specify "-" and "%" characters in the command line.

OPTIONS

`-add` *`vframe`* `-vtype` *`vtypename`* [*`ouvaluepath`*] [`-user` *`user@domain`* `-domaintype` *`domaintype`*]

Add a vframe *vframe* to the VCS One cluster. Do not use the word vframe to name a service group or virtual machine, VCS One reserves its use.

Use the `-vtype` *vtype* option to specify the vtype (the name of the virtualization technology) for the vframe. The accepted values for *vtype* follow:

- esxvm
- etc.

Specify the vtype using `-vtype` when creating the vframe.

`-delete [-force]` *`vframe`* [`-user` *`user@domain`* `-domaintype` *`domaintype`*]

Delete a vframe. If the vframe contains resources, the `-force` option can be used to delete the vframe along with its resources if all resources are offline.

`-move [-updateroles] [-refreshvars]` *`vframe(s)`* `-ou` *`ouvaluepath`* [`-user` *`user@domain`* `-domaintype` *`domaintype`*]

Move a vframe or vframes specified by *vframe(s)* to another node in the Organization Tree. If a user is assigned a role on the vframe and moving the vframe violates the rooted user rule, moving the vframe is not allowed. However, you can use `-updateroles` to forcibly move the vframe that updates the user's roles appropriately.

If you attempt to move a vframe and if the current value of its extended attributes (that are used as resource variables) changes at the new location, the move is rejected. To override this behavior and move the vframe, use `-refreshvars`. Doing so modifies the value of the resource attributes that use the variable.

`-compatible [-propagate] vframe1 vframe2 [-user user@domain -domaintype domaintype]`

Specify that *vframe1* is compatible with *vframe2*. If the command succeeds,*vframe2* is also compatible with *vframe1*.

If the two vframes are already compatible, the command reports this information in a message and makes no change.

When you define a vframe's compatibility with other vframes, the vframe's CompatibleVFrames and IncompatibleVFrames attributes are set. The CompatibleVFrames and IncompatibleVFrames attributes are mutually exclusive such that only one of the attributes may contain an explicit value. The other attribute contains a null value.

You can display the value of the CompatibleVFrames attribute using the command:

```
havframe -display vframe -attribute CompatibleVFrames
```

If a null value is shown, you can display the value of the IncompatibleVFrames attribute.

The command to define compatibility between one vframe and another does not replace the compatibility values previously defined for either of them. It modifies the sets of values for them. You cannot use the `havframe -modify` command to change the values of the CompatibleVFrames or IncompatibleVFrames attributes.

By default, all vframes are compatible with all other vframes. Compatible vframes may be online on the same pframe. The Policy Master brings vframes online on a pframe. During this process, the Policy Master first checks that the vframes are compatible with the vframes currently running on the pframe. The Policy Master typically attempts to relocate any lower priority incompatible vframes currently online on the pframe to another suitable, configured pframe. In the case of a manual online command, a user must use the `-ejectlowpri` option to attempt to relocate a low priority incompatible vframe.

When the vframes you specify are part of a `local` dependency, use the `-propagate` option or the command is rejected. The `-compatible -propagate` option applies to local and hard/firm/soft vframe dependencies.

Considerations when you use the `havframe -compatible` command include:

- You can define compatibility between only two vframes at one time, unless you specify a vframe is compatible with `ALLVFRAMES`. To set compatibility between one vframe and two others, run the `havframe -compatible` command twice. (Run the command once to set compatibility with the first vframe, and a second time to set compatibility with the second vframe.)
- Unless vframes are compatible with each other, they cannot form part of a local vframe dependency tree. Another precondition for vframes in a local vframe dependency tree is that each vframe must be compatible or incompatible with the same set of vframes. Use the `-propagate` option to set the compatibility for the entire vframe dependency tree.
- The command to specify compatibility fails if you issue it when either vframe is in transition. A vframe is in transition while it comes online or goes offline. The command succeeds for vframes intent to come online.
- The vframes you specify in the command must currently exist, and not be vframes you intend to add in the future.

`-displayea [`*`vframe(s)`*`] [-attribute `*`attribute(s)`*`] [-user `*`user@domain`*` -domaintype `*`domaintype`*`]`

Displays the extended attributes and their values for a specified vframe or vframe(s). If no vframe is specified, the extended attributes and values for all vframes are displayed.

`-compatible [-propagate] -setname `*`setname`*` -withsetname `*`setname`*` [-user `*`user@domain`*` -domaintype `*`domaintype`*`] [-info]`

Makes the set (*setname*) compatible with another set. If the command succeeds, the two sets are compatible.

If the two sets are already compatible, the command reports this information in a message and makes no change.

When the vframes are part of a `local` dependency, use the `-propagate` option. The `-compatible -propagate` option applies to `local` and `hard`/`firm`/`soft` vframe dependencies.

`-info` Displays the objects that the command acts upon when you issue it. When `-info` is specified, the command is not executed; only information is displayed.

`-compatible [-propagate] {-ou` *`ouexpression`* `| -ea` *`eaexpression`* `| -ou` *`ouexpression`* `-ea` *`eaexpression`*`} {-withou` *`ouexpression`* `| -withea` *`eaexpression`* `| -withou` *`ouexpression`* `-withea` *`eaexpression`*`} [-user` *`user@domain`* `-domaintype` *`domaintype`*`] [-info]`

Specify that the vframes that are included in an *ouexpression* and/or an *eaexpression* are compatible with the vframes included in another *ouexpression* and/or *eaexpression*. If the command succeeds, the vframes that are included in the second expression are also compatible with the first expression. If the two expressions have already been made compatible, the command reports this information in a message and makes no change.

When the vframes are part of a `local` dependency, use the `-propagate` option. The `-compatible -propagate` option applies to `local` and `hard`/`firm`/`soft` vframe dependencies.

`-info` Displays the objects that the command acts upon when you issue it. When `-info` is specified, the command is not executed; only information is displayed.

`-compatible [-propagate]` *`vframe`* `ALLVFRAMES [-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Specify that *vframe* is compatible with all other vframes in the VCS One cluster. If the command succeeds, all vframes are also compatible with *vframe*. Refer to the previous description on how to specify compatibility between two vframes.

`-incompatible [-propagate]` *`vframe1 vframe2`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Specify that *vframe1* is incompatible with *vframe2*. If the command succeeds, *vframe2* is also incompatible with *vframe1*. If you specify ALLVFRAMES, then a successful command reports *vframe2* to be incompatible with all vframes, and all vframes to be incompatible with *vframe2*.

If the two vframes are already incompatible, the command reports this information in a message and makes no change.

When you define a vframe's compatibility or incompatibility with other vframes, the vframe's CompatibleVFrames and IncompatibleVFrames attributes are set. The CompatibleVFrames and IncompatibleVFrames attributes are mutually exclusive such that only one of the attributes may contain an explicit value. The other attribute contains a null value.

You can display the value of the IncompatibleVFrames attribute using the command:

```
havframe -display vframe -attribute
IncompatibleVFrames
```

If a null value is shown, you can display the value of the CompatibleVFrames attribute.

The command to define incompatibility between one vframe and another does not replace the compatibility values previously defined for either of them. It modifies the sets of values for them. You cannot use the `havframe -modify` command to change the values of the CompatibleVFrames or IncompatibleVFrames attributes.

Incompatible vframes cannot be online on the same pframe. The Policy Master brings vframes online on a pframe. During this process, the Policy Master first checks the compatibility of the vframe with any vframes currently running on the pframe. The Policy Master attempts to relocate any lower priority incompatible vframes currently online on the pframe to another suitable, configured pframe. In the case of a manual online command, a user must use the `-ejectlowpri` option to attempt to relocate a low priority incompatible vframe.

When the vframes you specify are part of a local dependency, use the `-propagate` option or the command is rejected. The`-compatible -propagate` option applies to `local` and `hard`/`firm`/`soft` vframe dependencies.

When you run the `havframe -incompatible` command, keep in mind the following:

- You can define incompatibility between a vframe and only one other vframe at one time, unless you specify a vframe is incompatible with `ALLVFRAMES`. To set incompatibility between one vframe and two others, run the `havframe -incompatible` command twice. (Run the command once to set incompatibility with the first vframe, and a second time to set incompatibility with the second vframe.)
- Unless vframes are compatible with each other, they cannot form part of a local vframe dependency tree. Another precondition for vframes in a local vframe dependency tree is that each vframe must be compatible or incompatible with the same set of vframes. Use the `-propagate` option to set the compatibility for the entire vframe dependency tree.
- The command to specify incompatibility fails if you issue it when either vframe is in transition. A vframe is in transition while it comes online or goes offline. The command succeeds for vframes intent to come online.
- The vframes you specify in the command must currently exist, and not be vframes you intend to add in the future.

`-incompatible [-propagate] vframe ALLVFRAMES [-user user@domain -domaintype domaintype]`

Specify that *vframe* is incompatible with all other vframes in the VCS One cluster. If the command succeeds, all vframes are also incompatible with *vframe*. A vframe that is part of a local dependency tree cannot be made incompatible with `ALLVFRAMES`.

Refer to the previous description on how to specify incompatibility between two vframes.

`-incompatible [-propagate] -setname setname -withsetname setname [-user user@domain -domaintype domaintype] [-info]`

Specify that set specified by *setname* is incompatible with another set. If the command succeeds, the two sets are made incompatible.

If the two sets have already been made incompatible, the command reports the information in a message and makes no change.

When the vframes you specify are part of a local dependency, use the `-propagate` option. The `-compatible -propagate` option applies to `local` and `hard`/`firm`/`soft` vframe dependencies.

`-info` Displays the objects that the command acts upon when you issue it. When `-info` is specified, the command is not executed; only information is displayed.

`-incompatible [-propagate] {-ou ouexpression | -ea eaexpression | -ou ouexpression -ea eaexpression} {-withou ouexpression | -withea eaexpression | -withou ouexpression -withea eaexpression} [-user user@domain -domaintype domaintype] [-info]`

Specify that the vframes that are included in an *ouexpression* and/or an *eaexpression* are incompatible with the vframes included in another *ouexpression* and/or *eaexpression*. If the command succeeds, the vframes that are included in the second expression are made incompatible with the vframes included in the first expression.

If the two expressions have already been made incompatible, the command reports this information in a message and makes no change.

When the vframes you specify are part of a local dependency, use the `-propagate` option. The `-compatible -propagate` option applies to local and hard/firm/soft vframe dependencies.

`-info` Displays the objects that the command acts upon when you issue it. When `-info` is specified, the command is not executed; only information is displayed.

`-link` *`parentvframe childvframe | childgroup relationship`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Specify dependencies between vframes. The *childgroup* is name of the service group that is the child in the dependency. The variable *relationship* is one of the following:

`global [soft | firm | hard]`

When VCS One starts, the child vframe must be online on some system in the VCS One cluster before the parent vframe can be brought online.

With the dependency set to `soft`, if the child vframe faults and fails over, the parent vframe continues to remain online. If VCS One cannot bring the child vframe online in the VCS One cluster, the parent vframe remains online.

For a `firm` dependency, a parent vframe must be taken offline if its child vframe faults. When the child vframe fails over to another system, the parent can return online. If VCS One cannot bring the child vframe online in the VCS One cluster, the parent vframe remains offline.

For a `hard` dependency, the parents are taken offline before the child if the child vframe faults. If the child fails over, the parent fails over to another system. If the child cannot fail over, the parent remains offline. For a `hard` dependency, a child is taken offline if its parent faults. If the child fails over, the parent migrates to another system. If the child cannot fail over, the parent stays offline.

`local [soft | firm | hard]`

When VCS One starts, the child vframe must be online on the same system in the VCS One cluster before the parent vframe can be brought online.

For a `soft` dependency, the parent vframe continues to run on the local system if the child vframe faults. The parent runs on the local system until the child fails over to another system in the VCS One cluster. After the child fails over, the parent vframe fails over to the same system as the child. If VCS One cannot bring the child vframe online in the VCS One cluster, the parent vframe remains online.

With the dependency set to `firm`, if the child vframe faults, the parent vframe must go offline. If the child fails over, the parent vframe comes back online on the same system as the child. If VCS One cannot bring the child vframe online in the VCS One cluster, the parent vframe remains offline.

With the dependency set to `hard`, if the child vframe faults, the parents are taken offline before the child is taken offline. If the child fails over, the parent fails over to the same system. If the child cannot fail over, the

parent remains offline. With the dependency set to `hard`, if the parent faults, child is taken offline. If the child fails over, the parent migrates to the same system. If the child cannot fail over, the parent remains offline.

A vframe dependency tree may be at most five levels deep, and each parent can have only one child.

Parallel parent vframes dependent on parallel child vframes are not supported in global dependencies. The configuration of parallel parent vframes dependent on a failover child vframe is not supported in local dependencies.

`-link` *`parentgroup childvframe relationship`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Creates a dependency relationship between vframe and service group objects. The *parentgroup* is the name of the service group that is the parent (dependent) in the dependency.

`-unlink` *`parentvframe childvframe | childgroup`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Remove dependency between two vframes. Note that the dependency is not specified. The *childgroup* is name of the service group that is the child in the dependency.

`-unlink` *`parentgroup childvframe`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Removes the dependency relationship between vframe and service group objects. The *parentgroup* is the name of the service group that is the parent (dependent) in the dependency.

`-dep [`*`vframe(s)`*`] [-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Display dependencies between vframes.

`-linksys [-force]` *`vframe system`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Links a VCS One system to a vframe.

`-unlinksys` *`vframe system`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Removes a link from a VCS One system to a vframe.

`-associate` *`vframe objectname`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

`-associate` Builds associations between vframes and management servers (like VirtualCenter). The associations help with visualization of the server farm as well as enabling commands.

`-disassociate` *`vframe objectname`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

`-disassociate` Severs associations between vframes and command servers (like VirtualCenter).

`-clear [`*`vframe`* `| -setname` *`setname`* `| -ou` *`ouexpression`* `| -ea` *`eaexpression`* `| -ou` *`ouexpression`* `-ea` *`eaexpression`*`] [-pframe` *`pframe`*`] [-user` *`user@domain`* `-domaintype` *`domaintype`*`] [-info]`

Clear all faulted resources in the specified virtual machine, set, or virtual machines that are specified by `-ea` ***eaexpression*** and/or `-ou` ***ouexpression***, by changing their state from `faulted` to `offline`. If no pframe is specified, all resources are cleared on all pframes in the virtual machine's SystemList. A message is printed if no faulted resources exist.

`-clearadminwait [-fault]` *`vframe`* `-pframe` *`pframe`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Clear the `ADMIN_WAIT` state of all resources in the specified virtual machine on the specified pframe. If the resources continue in the `ADMIN_WAIT` state, use the `-fault` option to clear the `ADMIN_WAIT` state. The state of the resources is set to `ONLINE` | `UNABLE_TO_OFFLINE` or `FAULTED`, depending on the reasons the ResAdminWait trigger had been called.

Note that the `online`, `offline`, `switch`, and `flush` operations cannot be performed on resources in the ADMIN_WAIT state. Also, when resources are in the `ADMIN_WAIT` state, the `hastop` command requires the `-force` option.

`-flush [-action]` *`vframe`* `-pframe` *`pframe`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Flush a virtual machine and enable corrective action. All resources in the vframe waiting to come online automatically transition to not waiting. Resources waiting to go offline remain in that state. Any failovers and switches in progress are canceled.

`-action` Removes the vframe transition queue (GTQ) action entries for a vframe that is planned to be brought online or taken offline before you flush the vframe. If another vframe has a dependency on the planned online or offline action, the command fails. In this case, use either the `hagtq -abortaction` or `hagtq -aborttree` command instead. If the `-flush` option is used without the `-action` option for a vframe having planned GTQ online or offline action entries, the command fails.

`-flush [-intent]` *`vframe`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Flush all intent online entries for the specified vframe.

`-online [{-ejectlopri [-ignorestandby]} | -ignorestandby | -propagate] vframe -pframe pframe [-user user@domain -domaintype domaintype]`

Start a virtual machine (bring its resources online and power it on) on a specified pframe. By default, a resource's `AutoStart` attribute is set to `1`. If the `AutoStart` attribute is set to `0` for a resource, the command does not start that resource unless other resources with `AutoStart` set to `1` depend on that resource.

`-ejectlowpri` Specifies that lower priority vframes running on the specified pframe may be taken offline if they use the capacity that the specified vframe requires. Lower priority vframes can also be taken offline if they are incompatible with the specified vframe.

The `-propagate` option specifies that all of a vframe's required child vframes are brought online on the specified pframe if they are not currently online. In the following example, G1 depends on G2, and G2 depends on G3. When G1 is brought online with the `-propagate` option, G2 and G3 are brought online if they are not online already. The `-propagate` option applies for all child vframes, including those with `local`/`global hard`/`firm`/`soft` dependencies. The specified virtual machine must not be in transition. A virtual machine is in transition while it comes online, goes offline, or fails over to another pframe.

`-online [-ejectlowpri] [-nointent] vframe(s) -any [-user user@domain -domaintype domaintype]`

Start one or more specified virtual machines (bring their resources online and turn on their power) on the best possible pframe in a VCS One cluster.

By default, a resource's `AutoStart` attribute is set to `1`. If the `AutoStart` attribute is set to `0` for a resource, the command does not start that resource unless other resources with `AutoStart` set to `1` depend on that resource.

The `-ejectlowpri` option specifies that lower priority vframes running on the best possible available pframe may be taken offline if they use capacity required by a vframe being brought online or are incompatible with a vframe being brought online.

When a single vframe is specified, the command attempts to bring the vframe's child vframe online on an appropriate pframe if it is not currently online. Therefore, the attempt to online the vframe is not automatically rejected if the child is not already online.

If multiple vframes are specified, the command does not attempt to bring online any offline child vframes, in which case the command may not succeed.

Unless the -nointent option is used, the command adds vframes that cannot come online to the GTQ with "intentonline" entries.

The -any option will bring online a failover vframe on one pframe in the SystemList. For a parallel vframe, the -any option will bring online an additional instance of the vframe.

-online [-ejectlowpri] [-nointent] [-setname *setname* | -ou *ouexpression* | -ea *eaexpression* | -ou *ouexpression* -ea *eaexpression*] -any [-user *user@domain* -domaintype *domaintype*] [-info]

Start the virtual machines that are specified by a set name, or an *ouexpression* and/ or an *eaexpression* by bringing their resources online on the best possible pframe in a VCS One cluster.

The -online option can take either a set expression or an explicit list of objects as arguments.

Resources that have their AutoStart attribute set to zero (the default is one) are not started by this command unless resources that have AutoStart set to one depend on the resources.

The -ejectlowpri option specifies that lower priority vframes running on the best possible available pframe may be taken offline if they use capacity required by a vframe being brought online or are incompatible with a vframe being brought online.

When a single vframe is specified, the command attempts to bring the vframe's child vframe online on an appropriate pframe if it is not currently online. Therefore, the attempt to online the vframe is not automatically rejected if the child is not already online.

If multiple vframes are specified, the command does not attempt to bring online any offline child vframes, in which case the command may not succeed.

Unless the -nointent option is used, the command adds vframes that cannot come online to the GTQ with "intentonline" entries.

The -any option will bring online a failover vframe on one pframe in the SystemList.

Use the -info option to display the objects that the command will act upon if executed. When -info is specified, the command is not executed; only information is displayed.

`-online [-ejectlowpri] [-nointent] -all [-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Start a specified service group or multiple service groups by bringing their resources online on the best possible system in a VCS One cluster. For parallel groups, this option brings online all instances possible.

The `-ejectlowpri` option specifies that lower priority groups running on the best possible available system may be taken offline if they use capacity required by a group being brought online or are incompatible with a group being brought online.

Unless the `-nointent` option is used, the command adds groups that cannot come online to the GTQ with "intentonline" entries.

The `-everywhere` option applies only to a parallel group. It brings online a parallel service group on all systems in the SystemList.

`-offline [-propagate | -stopapps]` *`vframe`* `[-pframe` *`pframe`*`] [-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Stops a virtual machine and brings its resources offline on the specified pframe.

The `-propagate` option specifies that a vframe's `global`/`local` and `hard`/`firm` dependent parent vframes are brought offline if they are currently online. Parents with a `soft` dependency are not taken offline. For example, if G1 (on pframe A) has a `global firm` dependency on G2 (on pframe A), and G2 has a `global firm` dependency on G3 (on pframe B), then when the command to offline G3 is issued with the `-propagate` option, G1 and G2 are taken offline on pframe A and G3 is taken offline on pframe B.

When there are service group objects that are online on the vframe (the system linked to the vframe) then it is mandatory to specify the `stopapps` switch. The command will offline all the service groups that are online on the vframe.

`-offline [-propagate]` *`vframe`* `-everywhere [-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

The `-everywhere` option can be used to take a virtual machine and any dependent virtual machines offline on any pframes where they are online.

The `-propagate` option specifies that a vframe's `global`/`local` and `hard`/`firm` required child vframes are taken offline on the specified pframe if they are online. It does not apply for `soft` parent-child dependencies. For example, if G1 depends on G2, which depends on G3, when G1 is taken offline with the `-propagate` option, G2 and G3 are taken offline if they are not already offline.

Note that the specified virtual machine must not be currently in the process of coming online, going offline, or failing over to another pframe.

`-offline {-setname setname | -ou ouexpression | -ea eaexpression | -ou ouexpression -ea eaexpression} -everywhere [-info] [-user user@domain -domaintype domaintype]`

Stop a virtual machine or virtual machines specified by a setname or by an *ouexpression* and/or an *eaexpression* by taking their resources offline on any pframe within a VCS One cluster.

The `-offline` option can take either a set expression or an explicit list of objects as arguments.

The `-everywhere` option can be used to take a virtual machine and any dependent virtual machines offline on any pframes where they are online.

`-offline -force vframe -pframe pframe [-user user@domain -domaintype domaintype]`

Offline a vframe when a pframe is in the "daemon down, node active" (DDNA) state. To offline a vframe when a pframe is in the DDNA state, the vframe must not be in transition with respect to the pframe and remote resources must not be monitored by a control vframe.

`-switch [-ejectlowpri | -propagate] [-ignorestandby] vframe -to pframe [-user user@domain -domaintype domaintype]`

Switch a virtual machine from the pframe on which it is active to the specified pframe. The `-switch` option applies only to failover vframes (vframes that have the `Parallel` attribute set to zero).

The `-ejectlowpri` option specifies that lower priority vframes running on the specified pframe may be taken offline if they use capacity required by the specified vframe or are incompatible with the specified vframe.

With the `-propagate` option, the operation to switch a virtual machine propagates to all `global`/`local firm`/`hard` parents that are brought online on the same pframe specified. The operation does not apply to `soft` parent-child dependencies. The `-propagate` option will fail if a virtual machine has a `local soft` parent vframe online. It will succeed if there is a `global soft` parent vframe online. However, the switch will not be propagated to a `global soft` parent vframe.

The *vframe* to be switched using the `-propagate` option cannot be dependent on any child vframe.

For example: G1 depends on G2, which depends on G3. When G3 is switched from pframe 2 to pframe 3 with the `-propagate` option, G2 and G1 are taken offline and brought online on pframe 3. If G1 and G2 have global dependency

on G3 and are originally online on pframe 1, they are taken offline on pframe 1 and brought online on pframe 3.

Other limitations for switching vframes using the `-propagate` option include:

- The `-any` and `-ejectlowpri` options must not be specified.
- The parent vframe must not be in the vframe transition queue (GTQ) for bringing online or taking offline.
- The parent vframe cannot be parallel.
- Users must have privileges to operate all vframes switched.
- The vframes to be switched must not violate vframe dependencies or load limitations.
- The vframes to be switched must not have a local soft parent vframe online.

Use the `-info` option to display the objects that the command will act upon if executed. When `-info` is specified, the command is not executed; only information is displayed.

`-switch [-ejectlowpri] vframe -any [-user user@domain -domaintype domaintype]`

The `-any` option can be used to switch a virtual machine to the best possible pframe on which it is currently not online based on the value of the vframe's FailOverPolicy attribute.

The `-ejectlowpri` option specifies that lower priority vframes running on the best possible available pframe may be taken offline, if they use capacity required by a vframe being brought online or are incompatible with a vframe being brought online.

`-migrate [-ejectlowpri | -propagate] [-ignorestandby] vframe -to pframe [-user user@domain -domaintype domaintype]`

Some virtualization technologies support moving a running virtual machine from one pframe to another pframe without powering off the virtual machine. Use the `-migrate` option to initiate the migration process. Before you use the `-migrate` option, the vframe must be associated with the management console vobject to which it belongs.

For example, you have a virtual machine vm_01 that runs on pframe pf_01. You have another fail over pframe pf_02 on standby. The corresponding management console for the virtual machine is vc_01. Use the `-associate` option to associate the virtual machine with the management console vobject—which is typically a one-time operation.

```
havframe -associate vm_01 vc_01

havframe -migrate vm_01 -to pf_02
```

`-freeze [-propagate] vframe [-user user@domain -domaintype domaintype]`

Freeze a virtual machine (disables a vframe from coming online, going offline, and being failed over). The `-propagate` option must be used when freezing vframes, if the vframe dependency between child and parent vframes is `hard`. The `-freeze -propagate` option does not operate on `soft` parent-child dependencies.

`-unfreeze [-propagate] vframe [-user user@domain -domaintype domaintype]`

Unfreezes a virtual machine (re-enables a vframe to come online, go offline, and fail over). The `-propagate` option must be used when unfreezing vframes, if the vframe dependency between child and parent vframes is `hard`. The `-unfreeze -propagate` option does not operate on `soft` parent-child dependencies. It propagates the following attributes to immediate `hard` child vframes and `hard` parent vframes: `GrpFaultPolicy`, `NodeFaultPolicy`, `Evacuate`, and `Priority`.

`-enable vframe(s) [-pframe pframe] [-user user@domain -domaintype domaintype]`

Enables a virtual machine or vframes.

`-enable {-setname setname | -ou ouexpression | -ea eaexpression | -ou ouexpression -ea eaexpression} [-pframe pframe] [-user user@domain -domaintype domaintype]`

Enable virtual machines for the specified setname or *ouexpression* and/or *eaexpression*.

`-enable -all [-pframe pframe] [-user user@domain -domaintype domaintype]`

Enable all virtual machines.

`-disable vframe(s) [-pframe pframe] [-user user@domain -domaintype domaintype]`

Disable a virtual machine or vframes. Actions such as bringing vframes online or switching them are not permitted.

`-disable {-setname setname | -ou ouexpression | -ea eaexpression | -ou ouexpression -ea eaexpression} [-pframe pframe] [-user user@domain -domaintype domaintype]`

Disable virtual machines for the specified setname or *ouexpression* and/or *eaexpression*.

`-disable -all [-pframe pframe] [-user user@domain -domaintype domaintype]`

Disable all virtual machines.

`-enableresources vframe [-user user@domain -domaintype domaintype]`

Enable all resources in a virtual machine. Agents monitor the resources in the vframe.

`-disableresources vframe [-user user@domain -domaintype domaintype]`

Disable all resources in a virtual machine. Agents do not monitor the resources in the vframe.

`-resources vframe [-user user@domain -domaintype domaintype]`

Lists resources for a virtual machine.

`-changeload [-ejectlowpri | -tryswitch] vframe {key value} ... [-user user@domain -domaintype domaintype]`

Change the load value(s) for the specified virtual machine. Values are associated with the user-defined keys that specify a load component. Use the `havframe -display` command to display the current values. Note that the keys are used throughout the VCS One cluster and defined in the `PrecedenceOrder` assoc attribute.

When the virtual machine is already online or partially online, and the command increases the overall load component requirement to exceed the available capacity of the pframe, the command fails unless `-tryswitch` or`-ejectlowpri` is specified.

The `-ejectlowpri` option specifies that the Policy Master attempt to relocate lower priority virtual machine(s) to another suitable, configured pframe to allow current pframe capacity to support the new load requirement.

The `-tryswitch` option specifies that the Policy Master attempt to relocate lower priority virtual machine(s) to another suitable, configured pframe to allow the current pframe capacity to support the new load requirement. If the available capacity after the lower priority virtual machine(s) have been relocated is still insufficient, the command attempts to switch the vframe to another pframe that supports the load requirement. The switching of the specified virtual machine to another pframe may lead to relocating lower priority vframes from that pframe to another one.

If the attempts to increase the load of the specified vframe fails, the specified vframe continues with the original load value.

The Policy Master acts on the relocated vframes based on the value of their `GrpFaultPolicy` attribute. If the relocated vframes cannot be brought online

elsewhere, the Policy Master creates `intentonline` entries for them in the vframe transition queue (GTQ).

`-display [vframe(s) | -ou ouexpression | -ea eaexpression | -ou ouexpression -ea eaexpression | -setname setname] [-attribute attribute(s)] [-pframe pframe(s)] [-user user@domain -domaintype domaintype]`

Display the attributes and their values for a specified virtual machine or virtual machines specified by a setname or an *ouexpression* and/or an *eaexpression*. If no vframe is specified, the attributes and values for all vframes are displayed. If the pframe is specified, display the attributes and values for the specified vframe(s) on the specified pframe.

`-list [-vtype vtype] [conditional(s)] [-user user@domain -domaintype domaintype]`

Displays a list of vframes whose values match given conditional statement(s). Conditional statements can take three forms: Attribute=Value, Attribute!=Value, Attribute=~Value. Multiple conditional statements imply AND logic. If no conditional statement is specified, all vframes in the server farm are listed.

Use the `-vtype` option to display a list of vframes with a given vtype.

For example, `havframe -list -vtype esxvm` lists all the vframes that have the vtype of esxvm.

`-state [vframe(s) | -setname setname | -ou ouexpression | -ea eaexpression | -ou ouexpression -ea eaexpression] [-pframe pframe(s)] [-user user@domain -domaintype domaintype]`

Display the current state of the specified virtual machine or the virtual machines specified by a setname or an *ouexpression* and/or an *eaexpression* on the specified pframe(s).

`-value vframe attribute [-pframe pframe] [-user user@domain -domaintype domaintype]`

The `-value` option provides the value of a single vframe attribute.

For example, `havframe -value vframeX State -pframe pframeb` displays the value of the `State` attribute for the vframe `vframeX` on pframe `pframeb`. The pframe name must be specified for local attribute values, but not for global attribute values. The `-value` option is used instead of the `-display` option to display one specific attribute value rather than a table of many attribute values.

`-infovars` *`vframe attribute`* [*`key`*] [`-user` *`user@domain`* `-domaintype` *`domaintype`*]

Displays the resource attributes that use the specified attribute as a variable.

See EXAMPLES.

`-wait` *`vframe attribute value`* [`-pframe` *`pframe`*] [`-time` *`seconds`*] [`-user` *`user@domain`* `-domaintype` *`domaintype`*]

The `-wait` option is for use in scripts to direct the `havframe` command to wait until the value of the attribute has changed as specified, or until the number of seconds specified by *seconds* has elapsed. The *seconds* variable is an integer specifying seconds. If *seconds* is not specified, `havframe` waits indefinitely.

The `-wait` option can be used only with changes to scalar attributes.

The`-pframe` option can be applied only when the scope of the attribute is `local`.

See EXAMPLES.

`-addpframe [-propagate]` *`vframe pframe(s)`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*]

The `-addpframe` option adds a pframe to the SystemList of the specified vframe without having to specify the priority number for that new pframe. The Policy Master automatically assigns it the next available priority number.

`-modify` *`modify_options`*

The `-modify` option lets you modify a vframe's attributes. Some attributes, such as `ProbesPending`, are internal to VCS One and cannot be modified. You can modify any attribute that can be configured in main.xml.

The `-propagate` option must be used when modifying the `Priority`, `Evacuate`, `GrpFaultPolicy`, or `NodeFaultPolicy` attribute if the vframe dependency between child and parent vframes is `hard`. These attributes are propagated to immediate `hard` child vframes and `hard` parent vframes. They are not propagated for any `soft` parent-child dependencies.

The `-propagate` option must be used when modifying the `SystemList` or `SystemZones` attribute if the vframe dependency between the same priority child and parent vframes is `local` (this includes `hard`/`firm`/`soft` `local` vframe dependencies). The parent and child vframes must be the same type (that is, `parallel`/`parallel` or `failover`/`failover`).

You may modify a scalar attribute's existing value using only the `-modify` option.

To modify existing values for vector, keylist, or association attributes, one of the *modify_options* (which include `-add`, `-delete`, `-update`, and `-delete -keys`) is also required.

Refer to the following list of `-modify` commands. You may display the commands using `havframe -help -modify`.

`SCALAR`

```
havframe -modify [-propagate] vframe attribute value [-pframe
pframe]
```

`VECTOR`

Use the following command only when the attribute has no value:

```
havframe -modify [-propagate] vframe attribute value
...[-pframe pframe]
```

For vector attributes that have values defined, only the following operations are allowed:

```
havframe -modify [-propagate] vframe attribute -add value ...
[-pframe pframe]
```

```
havframe -modify [-propagate] vframe attribute -delete -keys
[-pframe pframe]
```

Note: You cannot delete an individual element of a VECTOR.

`KEYLIST`

Use the following command only when the attribute has no value:

```
havframe -modify [-propagate] vframe attribute key ...
[-pframe pframe]
```

For keylist attributes that have values defined, only the following operations are allowed.

```
havframe -modify [-propagate] vframe attribute -add key ...
[-pframe pframe]
```

```
havframe -modify [-propagate] vframe attribute -delete key
... [-pframe pframe]
```

```
havframe -modify [-propagate] vframe attribute -delete -keys
[-pframe pframe]
```

`ASSOCIATION`

Use the following command only when the attribute has no value:

```
havframe -modify [-propagate] vframe attribute {key value}
... [-pframe pframe]
```

For association attributes that have values defined, only the following operations are allowed. You cannot use `havframe -modify` to modify the values of a vframe's load components. Use the `-changeload` option.

```
havframe -modify [-propagate] vframe attribute -add {key
value}... [-pframe pframe]
```

```
havframe -modify [-propagate] vframe attribute - update {key
value}... [-pframe pframe]
```

```
havframe -modify [-propagate] vframe attribute - delete key...
[-pframe pframe]
```

```
havframe -modify [-propagate] vframe attribute - delete -keys
[-pframe pframe]
```

SPECIAL CASES

Use the following command only when the attribute has no value:

CASE 1

```
havframe -modify [-propagate] {vframe(s) | -ou expression |
-ea expression | -ou expression-ea expression | -setname
setname} SystemList -refresh [-user user@domain
-domaintypedomaintype]
```

This command modifies the SystemList attribute for specified vframes or vframes specified by a setname or an *ouexpression* and/or an *eaexpression*.

The SystemList will be populated with relevant pframes from the set specified by SystemListExpr.

For example, if 20 pframes are relevant and have the following vtypes: 10 solaris/sparc, 5 linux/x86, and 5 aix, and the vframe's vtype is linux/x86, then the command will populate `SystemList` with those 5 linux/x86 pframes.

An error is returned if SystemListExpr is not set.

CASE 2

```
havframe -modify vframe_name ContainerInfo - update Enabled
"0"
```

Before setting the `Enabled` attribute to 0 (Enabled=0), first delete the corresponding Project or Zone resource, otherwise, the state will be reported as UNKNOWN.

To remove the resource, enter:

```
hares -delete resource_name
```

Next, change the vframe's ContainerInfo: Enabled attribute to 0:

```
havframe -modify vframe_name ContainerInfo - update Enabled
"0"
```

Displays the version of `havframe`.

`-help [-modify | -link | -list]`

Displays usage for the `havframe` command. When you enter the command and an option without arguments, the syntax for the specific option displays.

The `-modify` option displays usage for the `modify` option. The `-link` option displays usage for the `link` option. The `-list` option displays usage for the `list` option.

`-version`

Displays the version of `havframe`.

EXAMPLES

To display the usage syntax for a specific command option, enter the command and an option without arguments. For example, enter:

```
# havframe -online
```

To bring vframe db_vm online on pframe esxbox1, enter:

```
# havframe -online db_vm -pframe esxbox1
```

Within a script, to direct the `havframe` command to wait until a scalar vframe level attribute is changed, enter:

```
# havframe -wait db_vm State ONLINE -pframe esxbox1
```

NOTES

The VCS One server may reject some `havframe` commands. For example, VCS One does not allow you to bring a failover virtual machine online on a pframe if the vframe is online elsewhere in the VCS One cluster, or if the vframe is faulted on that pframe.

When using the command to specify or modify an attribute's value that begins with a dash ("-"), precede the value with a percent sign ("%"). For example, specify `-y` as `%-y`. Likewise, precede a value that starts with a percent sign with another percent sign.

SEE ALSO

`hapframe`(1M),`hares`(1M),`haclus`(1M),`haconf`(1M),`halogin`(1M),`hagtq`(1M)

havobject

`havobject` – add, modify, delete, display, and list vobjects; and display the attribute value for a given vobject.

SYNOPSIS

```
UNIX: /opt/VRTSvcsone/bin/havobject

Windows: %VCSONE_HOME%\bin\havobject

havobject -add vobject vtype [ouvaluepath] [-user user@domain
-domaintype domaintype]
havobject -delete [-force] vobject [-user user@domain -domaintype
domaintype]
havobject -move [-updateroles] vobject -ou ouvaluepath [-user
user@domain -domaintype domaintype]
havobject -override vobject staticattribute [-user user@domain
-domaintype domaintype]
havobject -undo_override vobject staticattribute [-user user@domain
-domaintype domaintype]
havobject -display [vobject(s) | -ou ouexpression | -ea eaexpression
| -ou ouexpression -ea eaexpression | -setname setname] [-attribute
attribute(s)] [-vtype vtype(s)] [-user user@domain -domaintype
domaintype]
havobject -displayea [vobject(s)] [-attribute attribute(s)] [-user
user@domain -domaintype domaintype]
havobject -list [conditional(s)] [-user user@domain -domaintype
domaintype]
havobject -value vobject attribute [-user user@domain -domaintype
domaintype]
havobject -wait vobject attribute value [-time seconds] [-user
user@domain -domaintype domaintype]
havobject -modify modify_options
havobject [-help [-modify | -list]]
havobject -version
```

AVAILABILITY

VRTSvcsonec

DESCRIPTION

The `havobject` command administers vobjects in the VCS One cluster. A vobject is an object instance of a given vtype. A vtype is a virtual object-type definition that represents a single entity in a virtualization environment. A vobject is based on a vtype. After a vobject is created, it inherits the attributes and defaults of the vtype. A vobject can override all static attributes.

Use the `havobject` command to add, delete, display, or list vobjects. You can also use the `havobject` command to perform a specified action on a virtual machine and to display the attribute value for a given vobject.

A non-root user who has not run the `halogin` command can execute the `havobject` command using the `-user` *user@domain* option to execute the command with the privileges of the specified user. When issuing the command, the user must enter the fully qualified domain user name and supply a password when prompted. If necessary, the `-domaintype` option can specify the type of domain against which the user is to be authenticated. Supported domain types include:

"unixpwd"

"nt"

"nis"

"nisplus"

"ldap"

"pam"

"vx" (Symantec Private Domain)

The domain type, by default, is "vx". The domain type is case sensitive.

See NOTES for how to specify "-" and "%" characters in the command line.

OPTIONS

`-add` *`vobject vtype`* `[`*`ouvaluepath`*`] [-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Adds a vobject of the specified type.

`-delete` *`vobject`* `[-force] [-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Deletes a vobject from the configuration.

`-move [-updateroles] vobject(s) -ou ouvaluepath [-user user@domain -domaintype domaintype]`

Moves a vobject or vobjects specified by `-ou` *ouvaluepath* from one location in the organization tree to another. Use the `-updateroles` option to update the roles to reflect the change.

`-override vobject staticattribute [-user user@domain -domaintype domaintype]`

For a given vobject, permits a static vtype attribute to be overridden. After using this command, use the `-modify` option to modify the value. You can use the `-display` option to see the values of overridden attributes. The override attribute can be removed using the `-undo_override` option.

`-undo_override vobject staticattribute [-user user@domain -domaintype domaintype]`

Removes the overridden static attribute from the vobject's list of attributes.

`-display [vobject(s) | -ou ouexpression | -ea eaexpression | -ou ouexpression -ea eaexpression | -setname setname] [-attribute attribute(s)] [-vtype vtype(s)] [-user user@domain -domaintype domaintype]`

Displays the attributes and their values for the specified *vtype(s)*, *attribute(s)*, or *vobject(s)* specified by a *setname* or an *ouexpression* and/or *eaexpression*. Multiple options may be used. If no option is specified, attribute values for all vobjects are displayed, including overridden values.

Arguments for the `-ou` and `-ea` command options must be enclosed in double quotes if they contain spaces. For example:

```
havobject -display -ou "/lob=DCMG /ob=VCS" -attribute Capacity
```

An extended attribute value cannot contain a comma.

In addition, an extended attribute value or validation set cannot contain a single quote (') character. The single quote character serves as a delimiter for the value in an EA expression. However, single quotes can be used to enclose a multiword extended attribute value in an EA expression. For example:

```
havobject -display -ea "ea1= 'new value' and
ea2= 'new value2'"
```

`-displayea [vobject(s)] [-attribute attribute(s)] [-user user@domain -domaintype domaintype]`

Displays the extended attributes and their values for a specified vobject or vobjects. If no extended attribute is specified, the extended attributes and values for all vobjects are displayed.

`-list [conditional(s)] [-user user@domain -domaintype domaintype]`

Displays a list of the vobjects whose values match given conditional statements. Conditional statements can take three forms: Attribute=Value, Attribute!=Value, Attribute=~Value. Multiple conditional statements imply AND logic. If no conditional statement is specified, all vobjects are listed.

`-value vobject attribute [-user user@domain -domaintype domaintype]`

The `-value` option is used instead of the `-display` option when one specific attribute value is needed rather than a table of many attribute values.

For example, `havobject -value VC1 Username` displays the value of the `Username` attribute for the vobject `VC1`.

`-wait vobject attribute value [-time seconds] [-user user@domain -domaintype domaintype]`

The `-wait` option is for use in scripts to direct the `havobject` command to wait until the value of the attribute has changed as specified or until the duration specified by *seconds* has been reached. *seconds* is an integer specifying seconds. If seconds is not specified, `havobject` waits indefinitely.

The `-wait` option can be used only with changes to scalar attributes.

`-modify modify_options`

The `-modify` option lets you modify a vobject's attributes.

You may modify a scalar attribute's existing value.

You may not use `-modify` to change values already defined for a vector, a keylist, or an association attribute. For vector, keylist, and association attributes, use the *modify_options*, which include `-add`, `-delete`, `-update`, or`-delete -keys`.

Refer to the following list of `-modify` commands. You may display the commands using `havobject -help -modify`.

`SCALAR`

havobject -modify *vobject attribute value* [-user *user@domain* -domaintype *domaintype*]

`VECTOR`

Use the following command only when the attribute has no value:

```
havobject -modify vobject attribute value ... [-user
user@domain -domaintype domaintype]
```

For vector attributes that have values defined, only the following operations are allowed:

```
havobject -modify vobject attribute -add value... [-user
user@domain -domaintype domaintype]
```

```
havobject -modify vobject attribute -delete -keys [-user
user@domain -domaintype domaintype]
```

Note: You cannot delete an individual element of a VECTOR.

KEYLIST

Use the following command only when the attribute has no value:

```
havobject -modify vobject attribute key ... [-user user@domain
-domaintype domaintype]
```

For keylist attributes that have values defined, only the following operations are allowed.

```
havobject -modify vobject attribute -add key... [-user
user@domain -domaintype domaintype]
```

```
havobject -modify vobject attribute -delete key... [-user
user@domain -domaintype domaintype]
```

```
havobject -modify vobject attribute -delete -keys [-user
user@domain -domaintype domaintype]
```

ASSOCIATION

Use the following command only when the attribute has no value:

```
havobject -modify vobject attribute {key value} ... [-user
user@domain -domaintype domaintype]
```

For association attributes that have values defined, only the following operations are allowed.

```
havobject -modify vobject attribute -add {key value}... [-user
user@domain -domaintype domaintype]
```

```
havobject -modify vobject attribute -update {key value}...
[-user user@domain -domaintype domaintype]
```

```
havobject -modify vobject attribute -delete key... [-user
user@domain -domaintype domaintype]
```

```
havobject -modify vobject attribute -delete -keys [-user
user@domain -domaintype domaintype]
```

`[-help [-modify | -list]]`

Displays the command usage for `havobject`. The `-modify` option provides the usage for the `-modify` option and the `-list` option provides the usage for the `-list` option. When you enter `havobject -help` and an option without arguments, the syntax for the specified option displays.

`-version`

Displays the command version.

EXAMPLES

To display the usage syntax for a specified command option, enter the command option without arguments. For example, enter:

```
# havobject -value
```

NOTES

When using the command to specify or modify an attribute's value that begins with a dash ("-"), precede the value with a percent sign ("%"). For example, specify `-y` as `%-y`. Likewise, precede a value that starts with a percent sign with another percent sign.

hagetcf

`hagetcf` – gathers installed software, configuration, system logs, and related information and creates a gzip file, which Symantec Technical Support can use to troubleshoot VCS One issues

SYNOPSIS

```
UNIX: opt/VRTSvcsone/bin/hagetcf

Windows: %VCSONE_HOME%\bin\hagetcf

hagetcf [-s | -silent] [-d output_directory]
hagetcf [-help]
hagetcf [-version]
```

AVAILABILITY

VRTSvcsonec

DESCRIPTION

If you experience issues with VCS One, contact Symantec Technical Support for assistance. Symantec Technical Support may request that you run `hagetcf` and send them the generated gzip file so that they can analyze your VCS One cluster.

The `hagetcf` command gathers information about installed software, VCS One cluster configuration, systems, logs, and related information. It then creates a gzip file, which Symantec Technical Support can use to troubleshoot VCS One issues.

The output file size for the `hagetcf` command varies depending on the size of the log files and any core files that may be present. When choosing an output directory, avoid file systems with limited free space. Avoid saving `hagetcf` output to the root file system.

The `hagetcf` command gathers the following information:

- Installed software information
- System information
- Configuration information

If the system is part of the Policy Master cluster, this configuration information includes VCS, VCS One, and VCS One database information.

If the system is a VCS One client, this configuration information includes agent directory and agent framework information.

- Log information

If the system is part of the Policy Master cluster, this log information includes installation logs, VCS One logs, lock files, and VCS logs.

If the system is a VCS One client, this log information includes installation logs, log messages, and lock files.

- Important VCS One file information
- Symantec Product Authentication Service (AT) configuration and backup file information
- Web console information

The `hagetcf` command gathers sensitive information about your VCS One cluster environment. Set the proper file permissions on the gzip file and use a secure protocol when sending it to Symantec Technical Support.

You may run the `hagetcf` command in interactive or silent mode. By default, `hagetcf` runs in interactive mode and prompts you to specify an output directory for the gzip file.

You may save the gzip file to either the default `/var/tmp` directory or a specific directory.

Note: The `hagetcf` command options are different on Windows. For information on `hagetcf` on Windows, refer to the *Veritas Cluster Server One User's Guide*.

OPTIONS

`[-s|-silent][-d` *`output_directory`*`]`

Use the `-s` or `-silent` option to run the `hagetcf` command in silent mode. Use the `-d` ***output_ directory*** option to specify the desired output directory for the gzip file. If no output directory is specified, the default directory is `/var/tmp`.

`[-h|-help]`

Displays usage for the `hagetcf` command.

`[-version]`

Display the version of the command.

EXAMPLES

To run `hagetcf` in interactive mode, enter:

```
# hagetcf
```

At the prompt, specify the output directory for the gzip file. To run `hagetcf` in silent mode and use the default directory, enter:

```
# hagetcf -s
```

By default, the gzip file is saved in the `/var/tmp` directory. To run `hagetcf` in silent mode and specify a directory, enter:

```
# hagetcf -s -doutput_directory
```

SEE ALSO

`haconf`(1M)

hagrp

`hagrp` – perform VCS One service group operations

SYNOPSIS

UNIX: /opt/VRTSvcsone/bin/hagrp

Windows: %VCSONE_HOME%\bin\hagrp

hagrp -add *group* [-platform *platform*][*ouvaluepath*][-user *user@domain* -domaintype *domaintype*]
hagrp -delete [-force]*group*[-user *user@domain* -domaintype *domaintype*]
hagrp -move [-updateroles] [-refreshvars] *group(s)* -ou *ouvaluepath* [-user *user@domain* -domaintype *domaintype*]
hagrp -link *parentgroup childgroup* | *childvframe relationship* [-user *user@domain* -domaintype *domaintype*]
hagrp -link *parentvframe childgroup relationship* [-user *user@domain* -domaintype *domaintype*]
hagrp -unlink *parentgroup childgroup* | *childvframe* [-user *user@domain* -domaintype *domaintype*]
hagrp -unlink *parentvframe childgroup* [-user *user@domain* -domaintype *domaintype*]
hagrp -dep [*group(s)*] [-user *user@domain* -domaintype *domaintype*]
hagrp -clear {*group* | -setname *setname* | -ou *ouexpression* | -ea *eaexpression* | -ou *ouexpression* -ea *eaexpression*} [-sys *system*] [-user *user@domain* -domaintype *domaintype*]
hagrp -clearadminwait [-fault] *group* -sys *system* [-user *user@domain* -domaintype *domaintype*]
hagrp -flush [-action] *group* -sys *system* [-user *user@domain* -domaintype *domaintype*]
hagrp -flush [-intent] *group* [-user *user@domain* -domaintype *domaintype*]
hagrp -online [{-ejectlowpri [-ignorestandby]} | -ignorestandby | -propagate] *group* -sys *system* [-user *user@domain* -domaintype *domaintype*]
hagrp -online [-ejectlowpri] [-nointent] {*group(s)* -any | -all | *group* -everywhere} [-user *user@domain* -domaintype *domaintype*]
hagrp -online [-ejectlowpri] [-nointent] {-setname *setname* | -ou *ouexpression* | -ea *eaexpression* | -ou *ouexpression* -ea *eaexpression*} -any [-info] [-user *user@domain* -domaintype *domaintype*]

hagrp -offline [-offlinevframes] [-propagate] *group* [-sys *system*] [-user *user@domain* -domaintype *domaintype*]

hagrp -offline [-offlinevframes] [-propagate] *group* -everywhere [-user *user@domain* -domaintype *domaintype*]

hagrp -offline [-offlinevframes] {-setname *setname* | -ou *ouexpression* | -ea *eaexpression* | -ou *ouexpression* -ea *eaexpression*} -everywhere [-info] [-user *user@domain* -domaintype *domaintype*]

hagrp -offline -force *group* -sys *system* [-user *user@domain* -domaintype *domaintype*]

hagrp -switch [{-ejectlowpri [-ignorestandby]} | -ignorestandby | -propagate] *group* -to *system* [-user *user@domain* -domaintype *domaintype*]

hagrp -switch [-ejectlowpri] *group* -any [-user *user@domain* -domaintype *domaintype*]

hagrp -freeze [-propagate] *group* [-user *user@domain*-domaintype *domaintype*]

hagrp -unfreeze [-propagate] *group* [-user *user@domain* -domaintype *domaintype*]

hagrp -enable {*group(s)* | -setname *setname* | -ou *ouexpression* | -ea *eaexpression* | -ou *ouexpression* -ea *eaexpression*|-all} [-sys *system*] [-user *user@domain* -domaintype *domaintype*]

hagrp -disable {*group(s)* | -setname *setname* | -ou *ouexpression* | -ea *eaexpression* | -ou *ouexpression* -ea *eaexpression*|-all} [-sys *system*] [-user *user@domain* -domaintype *domaintype*]

hagrp -enableresources *group* [-user *user@domain* -domaintype *domaintype*]

hagrp -disableresources *group* [-user *user@domain* -domaintype *domaintype*]

hagrp -changeload [-ejectlowpri | -tryswitch] *group* {*key value*}... [-user *user@domain* -domaintype *domaintype*]

hagrp -display [*group(s)* | -ou *ouexpression* | -ea *eaexpression* | -ou *ouexpression* -ea *eaexpression* | -setname *setname*] [-attribute *attribute(s)*] [-sys *system(s)*] [-user *user@domain* -domaintype *domaintype*]

hagrp -displayea [*group(s)*] [-attribute *attribute(s)*] [-user *user@domain* -domaintype *domaintype*]

hagrp -list [*conditional(s)*] [-user *user@domain* -domaintype *domaintype*]

```
hagrp -state [group(s) | -setname setname | -ou ouexpression | -ea
eaexpression | -ou ouexpression -ea eaexpression ] [-sys system(s)]
[-user user@domain -domaintype domaintype]
hagrp -value group attribute [-sys system] [-user user@domain
-domaintype domaintype]
hagrp -resources group [-user user@domain -domaintype domaintype]
hagrp -infovars group attribute [key] [-user username@domain
-domaintype domaintype]
hagrp -wait group [-ea] attribute value [-sys {system|-any}] [-time
seconds] [-user user@domain -domaintype domaintype]
hagrp -addsystem [-propagate] group system(s) [-user user@domain
-domaintype domaintype]
hagrp -modify modify_options
hagrp -compatible options
hagrp -incompatible options
hagrp [-help [-modify | -compatible | -incompatible | -list]]
hagrp -version
```

AVAILABILITY

VRTSvcsonec

DESCRIPTION

A service group is an instance of an application service that is made highly available with VCS One. A service group comprises one or more resources of various resource types, such as disks, volumes, or databases. Use the `hagrp` command to manage service groups and to view information about them.

An OU expression cannot contain spaces.

An EA expression must be enclosed in double quotes if it contains spaces.

An extended attribute value cannot contain a comma.

In addition, an extended attribute value or validation set cannot contain a single quote (') character. The single quote character serves as a delimiter for the value in an EA expression. However, single quotes can be used to enclose an extended attribute value that has more than one word in an EA expression. For example:

```
hagrp -display -ea "ea1= 'new value' and ea2= 'new value2'"
```

For the `-platform` option, supported values for *platform* are:

- aix

- aix/rs6000 (alias aix)
- esx
- hpux
- linux
- linux/x86 (alias linux)
- solaris
- solaris/x86
- solaris/sparc (alias solaris)
- windows
- windows/x86

Use the explicit platform name where no alias is defined. When *platform* appears in any displays, the full name and not the alias is shown.

A non-root user who has not run the `halogin` command can execute the `hagrp` command using the `-user` *`user@domain`* option. This option executes the command with the privileges of the specified user. When you issue the command, enter your fully qualified domain user name and supply a password when prompted. If necessary, the `-domaintype` option can specify the type of domain against which the user is to be authenticated. Supported domain types include:

- "unixpwd"
- "nt"
- "nis"
- "nisplus"
- "ldap"
- "pam"
- "vx" (Symantec Private Domain)

The default domain type is "vx". The domain type is case sensitive.

See NOTES for how to specify "-" and "%" characters in the command line.

OPTIONS

`-add` *`group`* [`-platform` *`platform`*] [*`ouvaluepath`*] [`-user` *`user@domain`* `-domaintype` *`domaintype`*]

Add a service group to the VCS One cluster.

Use the `-platform` *`platform`* option to specify the platform for the group. The accepted values for *platform* are aix, aix/rs6000, esx, hpux, linux, linux/x86, solaris, solaris/x86, solaris/sparc, windows, and windows/x86. If you did not set a default platform for the VCS One cluster, specify the platform using `-platform` when you create the group. If you set a default platform for the VCS One cluster, it is used by default for a new group unless you specify the platform using `-platform`.

`-delete [-force]` *`group`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Delete a service group. To delete a service group that is part of a composite service group, first remove the service group from the composite service group's GroupList. To find the name of the composite service group that a service group belongs to, use the following command:

```
hagrp -value group csg_name
```

If the group contains resources, you can use the `-force` option to delete the group along with its resources if all resources are offline. To delete a service group that is part of a composite service group, first remove the service group from the composite service group's GroupList.

`-move [-updateroles] [-refreshvars]` *`group(s)`* `-ou` *`ouvaluepath`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Move the service group(s) that you specify using *group(s)* to another node in the Organization Tree. If a user is assigned a role on the service group and moving the group violates the rooted user rule, moving the group is not allowed. However, you can use `-updateroles` to forcibly move the group that updates the user's roles appropriately.

If you attempt to move a group and if the current value of any of its extended attributes (which is used as a resource variable) changes at the new location, the move is rejected. To override this behavior and move the system, use `-refreshvars`. Doing so will modify the value of the resource attributes that use the variable.

`-link` *`parentgroup childgroup | childvframe relationship`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Specify dependencies between service groups. The *childvframe* is the name of the vframe that is the child in the dependency. The variable *relationship* is one of the following:

`global [soft | firm | hard]`

When VCS One starts, the child group must be online on some system in the VCS One cluster before the parent group can be brought online.

With the dependency set to `soft`, if the child group faults and fails over, the parent group continues to remain online. If VCS One cannot bring the child group online in the VCS One cluster, the parent group remains online.

With the dependency set to `firm`, if the child group faults, the parent group must be taken offline until the child group fails over to another system, at which time the parent can return online. If VCS One cannot bring the child group online in the VCS One cluster, the parent group remains offline.

With the dependency set to `hard`, if the child group faults, the parents are taken offline before the child is taken offline. If the child fails over, the parent fails over to another system. If the child cannot fail over, the parent remains offline. With the dependency set to `hard`, if the parent faults, child is taken offline. If the child fails over, the parent migrates to another system. If the child cannot fail over, the parent remains offline.

`local [soft | firm | hard]`

When VCS One starts, the child group must be online on the same system in the VCS One cluster before the parent group can be brought online.

With the dependency set to `soft`, if the child group faults, the parent group continues to run on the local system until the child fails over to another system in the VCS One cluster, at which time the parent group will fail over to the same system as the child. If VCS One cannot bring the child group online in the VCS One cluster, the parent group remains online.

With the dependency set to `firm`, if the child group faults, the parent group must go offline. If the child fails over, the parent group comes back online on the same system as the child. If VCS One cannot bring the child group online in the VCS One cluster, the parent group remains offline.

With the dependency set to `hard`, if the child group faults, the parents are taken offline before the child is taken offline. If the child fails over, the parent fails over to the same system. If the child cannot fail over, the parent remains offline. With the dependency set to `hard`, if the parent faults, child is taken offline. If the child fails over, the parent migrates to the same system. If the child cannot fail over, the parent remains offline.

A group dependency tree may be at most five levels deep, and each parent can have only one child.

Parallel parent groups dependent on parallel child groups are not supported in global dependencies. The configuration of parallel parent groups dependent on a failover child group is not supported in local dependencies.

`-link` *`parentvframe childgroup relationship`* [`-user` *`user@domain`* `-domaintype` *`domaintype`*]

Creates a dependency between service group and vframe objects.

`-unlink` *`parentgroup childgroup | childvframe`* [`-user` *`user@domain`* `-domaintype` *`domaintype`*]

Removes a dependency between two service groups, or between a service group and a vframe. The dependency is not specified.

`-unlink` *`parentvframe childgroup`* [`-user` *`user@domain`* `-domaintype` *`domaintype`*]

Removes a dependency between two service groups, or between a service group and a vframe.

`-dep` [*`group(s)`*] [`-user` *`user@domain`* `-domaintype` *`domaintype`*]

Display dependencies between groups.

`-clear` {*`group`* | `-setname` *`setname`* | `-ou` *`ouexpression`* | `-ea` *`eaexpression`* | `-ou` *`ouexpression`* `-ea` *`eaexpression`*} [`-sys` *`system`*] [`-user` *`user@domain`* `-domaintype` *`domaintype`*]

Clear all faulted resources in the specified service group, set, or service groups specified by `-ea` *`eaexpression`* and/or `-ou` *`ouexpression`*, by changing their state from `faulted` to `offline`. If no system is specified, all resources are cleared on all systems in the group's SystemList. A message is printed if no faulted resources exist.

`-clearadminwait` [`-fault`] *`group`* `-sys` *`system`* [`-user` *`user@domain`* `-domaintype` *`domaintype`*]

Clear the `ADMIN_WAIT` state of all resources in the specified group on the specified system. If the resources continue in the `ADMIN_WAIT` state, use the `-fault` option to clear the `ADMIN_WAIT` state. The state of the resources is set to `ONLINE` | `UNABLE_TO_OFFLINE` or `FAULTED`, depending on the reasons the ResAdminWait trigger had been called.

Note that the `online`, `offline`, `switch`, and `flush` operations cannot be performed on resources in the `ADMIN_WAIT` state. Also, when resources are in the `ADMIN_WAIT` state, the `hastop` command requires the `-force` option.

`-flush [-action] group -sys system [-user user@domain -domaintype domaintype]`

Flush a service group and enable corrective action. All resources in the service group that are waiting to come online or go offline automatically transition to not waiting. Any failovers and switches in progress are cancelled.

The `-action` option removes the group transition queue (GTQ) action entries for a service group that is planned to be brought online or taken offline before it flushes that service group.

If you have the ROLE_FARM_GTQ privilege, `-action` cancels the actions of dependent service groups. If you do not have this privilege and another service group has a dependency on the planned online or offline action, the command fails. In this case, use either the `hagtq -abortaction` or `hagtq -aborttree` command instead. If the `-flush` option is used without the `-action` option for a service group that has planned GTQ online or offline action entries, the command fails.

`-flush [-intent] group [-user user@domain -domaintype domaintype]`

Flush all intent online entries in the GTQ for the specified service group.

`-online [{-ejectlowpri [-ignorestandby]} | -ignorestandby | -propagate] group -sys system [-user user@domain -domaintype domaintype]`

Start a service group by bringing its resources online on a specified system. Resources that have their `AutoStart` attribute set to zero (the default is one) are not started by this command unless resources that have `AutoStart` set to one depend on the resources.

The `-ejectlowpri` option specifies that lower priority groups running on the specified system may be taken offline if they use capacity required by the specified group or are incompatible with the specified group.

The `hagrp -online -sys` command cannot bring a Master Group online on a system where its Standby Group is not online. To bring a Master Group online on a system where its Standby Group is not online, use the `-ignorestandby` option.

The `-propagate` option specifies that all of a group's required child groups are brought online on the specified system if they are not currently online. For example, if G1 depends on G2, which depends on G3, when G1 is brought online with the `-propagate` option, G2 and G3 are brought online if they are not online. The `-propagate` option applies for all child groups, including those with `local`/`global hard`/`firm`/`soft` dependencies. Note that the specified service group must not be currently in the process of coming online, going offline, or failing over to another system.

`-online [-ejectlowpri] [-nointent] {`*`group(s)`* `-any | all |` *`group`* `-everywhere} [-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Start a specified service group or multiple service groups by bringing their resources online on the best possible system in a VCS One cluster. A parallel service group is brought online on multiple systems in a VCS One cluster if so configured. On each system, only a single instance of a parallel group is brought online.

Resources that have their `AutoStart` attribute set to zero (the default is one) are not started by this command unless resources that have `AutoStart` set to one depend on the resources.

The `-ejectlowpri` option specifies that lower priority groups running on the best possible available system may be taken offline if they use capacity required by a group being brought online or are incompatible with a group being brought online.

When a single group is specified, the command attempts to bring the group's child group online on an appropriate system if it is not currently online. Therefore, the attempt to online the group is not automatically rejected if the child is not already online.

If multiple groups are specified, the command does not attempt to bring online any offline child groups, in which case the command may not succeed.

Unless the `-nointent` option is used, the command adds groups that cannot come online to the GTQ with "intentonline" entries.

Use the `-any` option to bring online a failover group on one system in the SystemList. For a parallel group, the `-any` option will bring online an additional instance of the group.

Use the `-all` option to bring online all service groups under the user's privilege. This option brings all instances of a parallel service group online.

Use the `-everywhere` option to bring a parallel service group online on all systems in the SystemList. The `-everywhere` option applies only to a parallel service group.

`-online [-ejectlowpri] [-nointent] [-setname` *`setname`* `| -ou` *`ouexpression`* `| -ea` *`eaexpression`* `| -ou` *`ouexpression`* `-ea` *`eaexpression`*`] -any [-info] [-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Start the service groups specified by a set name, or an *ouexpression* and/or an *eaexpression* by bringing their resources online on the best possible system in a VCS One cluster. Parallel service groups are brought online on multiple systems in a VCS One cluster if so configured. On each system, only a single instance of a parallel group is brought online.

The -online option can take either a set expression or an explicit list of objects as arguments.

Resources that have their AutoStart attribute set to zero (the default is one) are not started by this command unless resources that have AutoStart set to one depend on the resources.

The -ejectlowpri option specifies that lower priority groups running on the best possible available system may be taken offline if they use capacity required by a group being brought online or are incompatible with a group being brought online.

When a single group is specified, the command attempts to bring the group's child group online on an appropriate system if it is not currently online. Therefore, the attempt to online the group is not automatically rejected if the child is not already online.

If multiple groups are specified, the command does not attempt to bring online any offline child groups, in which case the command may not succeed.

Unless the -nointent option is used, the command adds groups that cannot come online to the GTQ with "intentonline" entries.

The -any option will bring online a failover group on one system in the SystemList. For a parallel group, the -any option will bring online an additional instance of the group.

Use the -info option to display the objects that the command will act upon if executed. When -info is specified, the command is not executed; only information is displayed.

`-offline [-offlinevframes] [-propagate] group [-sys system] [-user user@domain -domaintype domaintype]`

Stop a service group by taking its resources offline on the specified system.

If the service group runs on a system linked to a vframe, -offlinevframes takes the vframe offline.

The -propagate option specifies that a group's global/local and hard/firm dependent parent groups are brought offline if they are currently online. Parents with a soft dependency are not taken offline. For example, if G1 (on system A) has a global firm dependency on G2 (on system A), and G2 has a global firm dependency on G3 (on system B), then when the command to offline G3 is issued with the -propagate option, G1 and G2 are taken offline on system A and G3 is taken offline on system B.

`-offline [-offlinevframes] [-propagate] group -everywhere [-user user@domain -domaintype domaintype]`

If the service group runs on a system linked to a vframe, `-offlinevframes` takes the vframe offline.

The `-everywhere` option can be used to take a service group and any dependent service groups offline on any systems where they are online.

The `-propagate` option specifies that a group's `global`/`local` and `hard`/`firm` required child groups are taken offline on the specified system if they are online. It does not apply for `soft` parent-child dependencies. For example, if G1 depends on G2, which depends on G3, when G1 is taken offline with the `-propagate` option, G2 and G3 are taken offline if they are not already offline. Note that the specified service group must not be currently in the process of coming online, going offline, or failing over to another system.

`-offline [-offlinevframes] {-setname setname | -ou ouexpression | -ea eaexpression | -ou ouexpression -ea eaexpression} -everywhere [-info] [-user user@domain -domaintype domaintype]`

Stop a service group or service groups specified by a setname or by an *ouexpression* and/or an *eaexpression* by taking their resources offline on any system within a VCS One cluster.

The `-offline` option can take either a set expression or an explicit list of objects as arguments.

If the service group runs on a system linked to a vframe, `-offlinevframes` takes the vframe offline.

The `-everywhere` option can be used to take a service group and any dependent service groups offline on any systems where they are online.

`-offline -force group -sys system [-user user@domain -domaintype domaintype]`

Offline a group when a system is in the "daemon down, node active" (DDNA) state. To offline a group when a system is in the DDNA state, the group must not be in transition with respect to the system and remote resources must not be monitored by a control group.

`-switch [{-ejectlowpri [-ignorestandby]} -ignorestandby | -propagate] group -to system [-user user@domain -domaintype domaintype]`

Switch a service group from the system on which it is active to the specified system. The `-switch` option applies only to failover groups (groups that have the `Parallel` attribute set to zero).

The `hagrp -switch -to` command cannot switch a Master Group on a system where its Standby Group is not online. To switch a Master Group to a system where its Standby Group is not online, use the `-ignorestandby` option.

The `-ejectlowpri` option specifies that lower priority groups running on the specified system may be taken offline if they use capacity required by the specified group or are incompatible with the specified group.

With the `-propagate` option, the operation to switch a service group propagates to all `global`/`local firm`/`hard` parents that are brought online on the same system specified. The operation does not apply to `soft` parent-child dependencies. The `-propagate` option will fail if a service group has a `local soft` parent group online. It will succeed if there is a `global soft` parent group online. However, the switch will not be propagated to a `global soft` parent group.

The *group* to be switched using the `-propagate` option cannot be dependent on any child group.

For example: G1 depends on G2, which depends on G3. When G3 is switched from system 2 to system 3 with the `-propagate` option, G2 and G1 are taken offline and brought online on system 3. If G1 and G2 have global dependency on G3 and are originally online on system 1, they are taken offline on system 1 and brought online on system 3.

Other limitations for switching groups using the `-propagate` option include:

- The `-any` and `-ejectlowpri` options must not be specified.
- The parent group must not be in the group transition queue (GTQ) for taking online or offline.
- The parent group cannot be parallel.
- Users must have privileges to operate all groups switched.
- The groups to be switched must not violate group dependencies or load limitations.
- The groups to be switched must not have a local soft parent group online.

Use the `-info` option to display the objects that the command will act upon if executed. When `-info` is specified, the command is not executed; only information is displayed.

`-switch [-ejectlowpri] group -any [-user user@domain -domaintype domaintype]`

The `-any` option can be used to switch a service group to the best possible system on which it is currently not online based on the value of the group's `FailOverPolicy` attribute.

The `-ejectlowpri` option specifies that lower priority groups running on the best possible available system may be taken offline, if they use capacity required by a group being brought online or are incompatible with a group being brought online.

`-freeze [-propagate] group [-user user@domain -domaintype domaintype]`

Freeze a service group (disable groups from coming online, going offline, and being failed over). The `-propagate` option must be used when freezing groups, if the group dependency between child and parent groups is `hard`. The `-freeze -propagate` option does not operate on `soft` parent-child dependencies.

`-unfreeze [-propagate] group [-user user@domain -domaintype domaintype]`

Unfreezes a service group (that is, re-enables groups to come online, go offline, and fail over). The `-propagate` option must be used when unfreezing groups, if the group dependency between child and parent groups is `hard`. The `-unfreeze -propagate` option does not operate on `soft` parent-child dependencies. It propagates the following attributes to immediate `hard` child groups and `hard` parent groups: `GrpFaultPolicy`, `NodeFaultPolicy`, `Evacuate`, and `Priority`.

`-enable {group(s)| -setname setname | -ou ouexpression | -ea eaexpression | -ou ouexpression -ea eaexpression |-all} [-sys system] [-user user@domain -domaintype domaintype]`

Enable service groups for the specified service group(s), setname, or *ouexpression* and/or *eaexpression*.

Use the `-all` option to enable all service groups.

`-disable {group(s)| -setname setname | -ou ouexpression | -ea eaexpression | -ou ouexpression -ea eaexpression |-all} [-sys system] [-user user@domain -domaintype domaintype]`

Disable service groups for the specified service group(s), setname, or *ouexpression* and/or *eaexpression*. Actions such as bringing service groups online or switching them are not permitted.

Use the `-all` option to disable all service groups.

`-enableresources` *group* [`-user` *user@domain* `-domaintype` *domaintype*]

Enable all resources in a service group. Agents monitor the resources in the group.

`-disableresources` *group* [`-user` *user@domain* `-domaintype` *domaintype*]

Disable all resources in a service group. Agents do not monitor the resources in the group.

`-changeload [-ejectlowpri | -tryswitch]` *group* {*key value*}… [`-user` *user@domain* `-domaintype` *domaintype*]

Change the load value(s) for the specified service group. Values are associated with the user-defined keys that specify a load component. Use the `hagrp -display` command to display the current values. Note that the keys are used throughout the VCS One cluster and defined in the `PrecedenceOrder` assoc attribute for the VCS One cluster.

When the service group is already online or partially online, and the command increases the overall load component requirement to exceed the available capacity of the system, the command fails unless `-tryswitch` or `-ejectlowpri` is specified.

`-ejectlowpri` Directs the Policy Master to attempt to relocate lower priority service group(s) to another suitable, configured system to allow current system capacity to support the new load requirement. If it cannot relocate the service group(s), `-changeload` is rejected.

`-tryswitch` Directs the Policy Master to attempt to relocate lower priority service group(s) to another suitable, configured system to allow the current system capacity to support the new load requirement. If the available capacity after the lower priority service group(s) have been relocated is still insufficient, the command attempts to switch the group to another system that supports the load requirement. The switching of the specified service group to another system may lead to relocating lower priority groups from that system to another one.

If the attempts to increase the load of the specified group fails, the specified group continues with the original load value.

The Policy Master acts on the relocated groups based on the value of their `GrpFaultPolicy` attribute. If the relocated groups cannot be brought online elsewhere, the Policy Master creates `intentonline` entries for them in the group transition queue (GTQ).

`-display [group(s) | -ou ouexpression | -ea eaexpression | -ou ouexpression -ea eaexpression | -setname setname] [-attribute attribute(s)] [-sys system(s)] [-user user@domain -domaintype domaintype]`

Display the attributes and their values for a specified service group or service groups specified by a setname or an *ouexpression* and/or an *eaexpression*. If no group is specified, the attributes and values for all groups are displayed. If the system is specified, display the attributes and values for the specified group(s) on the specified system.

`-displayea [group(s)] [-attribute attribute(s)] [-user user@domain -domaintype domaintype]`

Display the extended attributes and their values for a specified group or groups. If no extended attribute is specified, the extended attributes and values for all groups are displayed.

`-list [conditional(s)] [-user user@domain -domaintype domaintype]`

Displays a list of groups whose values match given conditional statement(s). Conditional statements can take three forms: Attribute=Value, Attribute!=Value, Attribute=~Value. Multiple conditional statements imply AND logic. If no conditional statement is specified, all groups in the VCS One cluster are listed.

`-state [group(s) | -setname setname | -ou ouexpression | -ea eaexpression | -ou ouexpression -ea eaexpression] [-sys system(s)] [-user user@domain -domaintype domaintype]`

Display the current state of the specified service group or the service groups specified by a setname or an *ouexpression* and/or an *eaexpression* on the specified system(s).

`-value group attribute [-sys system] [-user user@domain -domaintype domaintype]`

The `-value` option provides the value of a single group attribute.

For example, `hagrp -value groupX State -sys sysb` displays the value of the `State` attribute for the group `groupX` on system `sysb`. The system name must be specified for local attribute values, but not for global attribute values. The `-value` option is used instead of the `-display` option to display one specific attribute value rather than a table of many attribute values.

`-infovars group attribute [key] [-user username@domain -domaintype domaintype]`

Displays the resource attributes that use the specified attribute as a variable. See EXAMPLES.

`-resources` *group* [`-user` *username@domain* `-domaintype` *domaintype*]

Lists resources for a service group.

`-wait` *group* [`-ea`] *attribute value* [`-sys` {*system*| `-any`}] [`-time` *seconds*] [`-user` *user@domain* `-domaintype` *domaintype*]

The `-wait` option is for use in scripts to direct the `hagrp` command to wait until the value of the attribute changes to the specified value, or until the number of seconds specified by *seconds* is reached. The *seconds* variable is an integer specifying seconds. If *seconds* is not specified, `hagrp` waits indefinitely.

Use the `-ea` option to direct the `hagrp` command to wait until the value of an extended attribute changes to the specified value.

The `-wait` option can be used only with changes to scalar attributes. The `-sys` option can be applied only when the scope of the attribute is `local`.

See EXAMPLES.

`-addsystem` [`-propagate`] *group system(s)* [`-user` *user@domain* `-domaintype` *domaintype*]

The `-addsystem` option adds a system to the SystemList of the specified group without having to specify the priority number for that new system. The Policy Master automatically assigns it the next available priority number.

`-modify` *modify_options*

The `-modify` option lets you modify a service group's attributes. Some attributes, such as `ProbesPending`, are internal to VCS One and cannot be modified. You can modify any attribute that can be configured in main.xml.

The `-propagate` option must be used when modifying the `Priority`, `Evacuate`, `GrpFaultPolicy`, or `NodeFaultPolicy` attribute if the group dependency between child and parent groups is `hard`. These attributes are propagated to immediate `hard` child groups and `hard` parent groups. They are not propagated for any `soft` parent-child dependencies.

The `-propagate` option must be used when modifying the `SystemList` or `SystemZones` attribute if the group dependency between the same priority child and parent groups is `local` (this includes `hard`/`firm`/`soft local` group dependencies). The parent and child groups must be the same type (that is, `parallel`/`parallel` or `failover`/`failover`).

You may modify a scalar attribute's existing value using only the `-modify` option.

To modify existing values for vector, keylist, or association attributes, one of the *modify_options* (which include `-add`, `-delete`, `-update`, and `-delete -keys`) is also required.

Refer to the following list of `-modify` commands. You may display the commands using `hagrp -help -modify`.

`SCALAR`

`hagrp -modify [refreshvars][-propagate]group attribute value -sys system]`

If you attempt to modify an extended attribute value that is a variable, an error message is displayed and the value is not modified. To override this behavior and modify an extended attribute that is a variable, use the `-refreshvars` option. Doing so will modify the value of the resource attributes that use the variable.

`VECTOR`

Use the following command only when the attribute has no value:

`hagrp -modify [-propagate] group attribute value... [-sys system]`

For vector attributes that have values defined, only the following operations are allowed:

`hagrp -modify [-propagate] group attribute -add value...[-sys system]`

`hagrp -modify [-propagate] group attribute -delete -keys [-sys system]`

Note: You cannot delete an individual element of a VECTOR.

`KEYLIST`

Use the following command only when the attribute has no value:

`hagrp -modify [-propagate] group attribute key… [-sys system]`

For keylist attributes that have values defined, only the following operations are allowed.

`hagrp -modify [-propagate] group attribute -add key...[-sys system]`

`hagrp -modify [-propagate] group attribute -delete key...[-sys system]`

```
hagrp -modify [-propagate] group attribute -delete -keys [-sys
system]
```

ASSOCIATION

Use the following command only when the attribute has no value:

```
hagrp -modify [-propagate] group attribute {key value}...
[-sys system]
```

For association attributes that have values defined, only the following operations are allowed.

Note: You cannot use `hagrp -modify` to modify the values of a service group's load components. You must use the `-changeload` option.

```
hagrp -modify [-propagate] group attribute -add {key
value}...[-sys system]
```

```
hagrp -modify [-propagate] group attribute -update {key
value}...[-sys system]
```

```
hagrp -modify [-propagate] group attribute -delete key...[-sys
system]
```

```
hagrp -modify [-propagate] group attribute -delete -keys [-sys
system]
```

SPECIAL CASES

CASE 1

```
hagrp -modify [-propagate] {group(s) | -ou expression | -ea
expression | -ou expression -ea expression | -setname setname}
SystemList -refresh [-user user@domain -domaintype domaintype]
```

This command modifies the SystemList attribute for specified service groups or service groups specified by a setname or an *ouexpression* and/or an *eaexpression*.

The SystemList will be populated with relevant systems from the set specified by SystemListExpr.

For example, if 20 systems are relevant and have the following platforms: 10 solaris/sparc, 5 linux/x86, and 5 aix, and the group's platform is linux/x86, then the command will populate `SystemList` with those 5 linux/x86 systems.

An error is returned if SystemListExpr is not set.

CASE 2

```
hagrp -modify sg_name ContainerInfo -update Enabled "0"
```

Before setting the `Enabled` attribute to 0 (`Enabled`=0), you must first delete the corresponding Project or Zone resource, otherwise, the state will be reported as UNKNOWN.

To remove the resource, enter:

```
hares -delete resource_name
```

Next, change the Service Group's ContainerInfo: Enabled attribute to 0:

```
hagrp -modify sg_name ContainerInfo -update Enabled "0"
```

`-compatible [-propagate] group1 group2 [-user user@domain -domaintype domaintype]`

Specify that *group1* is compatible with *group2*. If the command succeeds, *group2* is also compatible with *group1*.

If the two groups are already compatible, the command reports this information in a message and makes no change.

When you define a service group's compatibility with other groups, the service group's CompatibleGroups and IncompatibleGroups attributes are set. The CompatibleGroups and IncompatibleGroups attributes are mutually exclusive such that only one of the attributes may contain an explicit value. The other attribute contains a null value.

You can display the value of the CompatibleGroups attribute using the command:

```
hagrp -display group -attribute CompatibleGroups
```

If a null value is shown, you can display the value of the IncompatibleGroups attribute.

The command to define compatibility between one group and another does not replace the compatibility values previously defined for either of them, but modifies the sets of values for them. You cannot use the `hagrp -modify` command to change the values of the CompatibleGroups or IncompatibleGroups attributes.

By default, all groups are compatible with all other groups. Compatible groups may be online on the same system. When the Policy Master attempts to bring a service group online on a system, it checks for the compatibility of the group with any groups currently running on the system. The Policy Master typically attempts to relocate any lower priority incompatible groups currently online on the system to another suitable, configured system. In the case of a manual

online command, a user must use the `-ejectlowpri` option to attempt to relocate a low priority incompatible group.

When the service groups you specify are part of a `local` dependency, you must use the `-propagate` option or else the command is rejected. The `-compatible -propagate` option applies to `local` and `hard`/`firm`/`soft` group dependencies.

Considerations when using the `hagrp -compatible` command include:

- You can define compatibility between only two groups at one time, unless you specify a group is compatible with `ALLGROUPS`. To set compatibility between one group and two others, run the `hagrp -compatible` command twice. (Run the command once to set compatibility with the first group, and a second time to set compatibility with the second group.)
- Unless groups are compatible with each other, they cannot form part of a local group dependency tree. Another precondition for groups in a local group dependency tree is that each group must be compatible or incompatible with the same set of service groups. Use the `-propagate` option to set the compatibility for the entire group dependency tree.
- The command to specify compatibility fails if it is issued when either group is in transition, that is, coming online or going offline. The command succeeds for groups intent to come online.
- The groups specified in the command must currently exist, and not be groups you intend to add in the future.

`-compatible [-propagate] group ALLGROUPS [-user user@domain -domaintype domaintype]`

Specify that *group* is compatible with all other groups in the VCS One cluster. If the command succeeds, all groups are also compatible with *group*.

Refer to the description for specifying compatibility between two groups above for additional information on specifying compatibility.

`-compatible [-propagate] -setname setname -withsetname setname [-info] [-user user@domain -domaintype domaintype]`

Specify that a set specified by *setname* is compatible with another set. If the command succeeds, the two sets are compatible.

If the two sets have already been made compatible, the command reports this information in a message and makes no change.

When the service groups are part of a `local` dependency, use the `-propagate` option. The `-compatible -propagate` option applies to `local` and `hard`/`firm`/`soft` group dependencies.

Use the `-info` option to display the objects that the command will act upon if executed. When `-info` is specified, the command is not executed; only information is displayed.

`-compatible [-propagate] {-ou` *`ouexpression`* `| -ea` *`eaexpression`* `| -ou` *`ouexpression`* `-ea` *`eaexpression`*`} {-withou` *`ouexpression`* `| -withea` *`eaexpression`* `| -withou` *`ouexpression`* `-withea` *`eaexpression`*`} [-info] [-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Specify that the groups included in an *ouexpression* and/or *eaexpression* are compatible with the groups included in another *ouexpression* and/or *eaexpression*. If the command succeeds, the groups included in the second expression are also compatible with the first expression. If the two expressions have already been made compatible, the command reports this information in a message and makes no change.

When the service groups are part of a `local` dependency, use the `-propagate` option. The `-compatible -propagate` option applies to `local` and `hard`/`firm`/`soft` group dependencies.

Use the `-info` option to display the objects that the command will act upon if executed. When `-info` is specified, the command is not executed; only information is displayed.

`-incompatible [-propagate]` *`group1 group2`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Specify that *group1* is incompatible with *group2*.

If the command succeeds, *group2* or all groups, if specified, are also incompatible with *group1*.

If the two groups are already incompatible, the command reports this information in a message and makes no change.

When you define a service group's compatibility or incompatibility with other groups, the service group's CompatibleGroups and IncompatibleGroups attributes are set. The CompatibleGroups and IncompatibleGroups attributes are mutually exclusive such that only one of the attributes may contain an explicit value. The other attribute contains a null value.

You can display the value of the IncompatibleGroups attribute using the command:

```
hagrp -display group -attribute IncompatibleGroups
```

If a null value is shown, you can display the value of the `CompatibleGroups` attribute.

The command to define incompatibility between one group and another does not replace the compatibility values previously defined for either of them, but modifies the sets of values for them. You cannot use the `hagrp -modify` command to change the values of the CompatibleGroups or IncompatibleGroups attributes.

Incompatible groups cannot be online on the same system. When the Policy Master attempts to bring a service group online on a system, it checks for the compatibility of the group with any groups currently running on the system. The Policy Master attempts to relocate any lower priority incompatible groups currently online on the system to another suitable, configured system. In the case of manual online command, a user must use the `-ejectlowpri` option to attempt to relocate a low priority incompatible group.

When the service groups you specify are part of a local dependency, you must use the `-propagate` option or else the command is rejected. The `-compatible -propagate` option applies to `local` and `hard`/`firm`/`soft` group dependencies.

Considerations when using the `hagrp -incompatible` command include:

- You can define incompatibility between a group and only one other group at one time, unless you specify a group is incompatible with `ALLGROUPS`. To set incompatibility between one group and two others, run the `hagrp -incompatible` command twice. (Run the command once to set incompatibility with the first group, and a second time to set incompatibility with the second group.)
- Unless groups are compatible with each other, they cannot form part of a local group dependency tree. Another precondition for groups in a local group dependency tree is that each group must be compatible or incompatible with the same set of service groups. Use the `-propagate` option to set the compatibility for the entire group dependency tree.
- The command to specify incompatibility fails if it is issued when either group is in transition, that is, coming online or going offline. The command succeeds for groups intent to come online.
- The groups specified in the command must currently exist, and not be groups you intend to add in the future.

`-incompatible [-propagate]` *`group`* `ALLGROUPS [-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Specify that *group* is incompatible with all other groups in the VCS One cluster. If the command succeeds, all groups are also incompatible with *group*. A group that is part of a local dependency tree cannot be made incompatible with `ALLGROUPS`.

Please refer to the description for specifying incompatibility between two groups above for additional information on specifying incompatibility.

`-incompatible [-propagate] -setname` *`setname`* `-withsetname` *`setname`*`[-info] [-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Specify that set specified by *setname* is incompatible with another set. If the command succeeds, the two sets are made incompatible.

If the two sets have already been made incompatible, the command reports the information in a message and makes no change.

When the service groups you specify are part of a local dependency, use the `-propagate` option. The `-compatible -propagate` option applies to `local` and `hard`/`firm`/`soft` group dependencies.

Use the `-info` option to display the objects that the command will act upon if executed. When `-info` is specified, the command is not executed; only information is displayed.

`-incompatible [-propagate] {-ou` *`ouexpression`* `| -ea` *`eaexpression`* `| -ou` *`ouexpression`* `-ea` *`eaexpression`*`} {-withou` *`ouexpression`* `| -withea` *`eaexpression`* `| -withou` *`ouexpression`* `-withea` *`eaexpression`*`} [-info] [-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Specify that the groups included in an *ouexpression* and/or an *eaexpression* are incompatible with the groups included in another *ouexpression* and/or *eaexpression*. If the command succeeds, the groups included in the second expression are made incompatible with the groups included in the first expression.

If the two expressions have already been made incompatible, the command reports this information in a message and makes no change.

`-help [-modify | -compatible | -incompatible | -list]`

Displays usage for the `hagrp` command. When you enter the command and an option without arguments, the syntax for the specific option displays.

The `-modify` option displays usage for the `modify` option. The `-compatible` option displays usage for the `compatible` option. The `-incompatible` option displays usage for the `incompatible` option. The `-list` option displays usage for the `list` option.

`-version`

Displays the version of `hagrp`.

EXAMPLES

Example 1. To display the usage syntax for a specific command option, enter the command and an option without arguments. For example, enter:

```
# hagrp -online
```

Example 2. To bring group `db_grp` online on system `mars01`, enter:

```
# hagrp -online db_grp -sys mars01
```

Example 3. Within a script, to direct the `hagrp` command to wait until a scalar group level attribute is changed, enter:

```
# hagrp -wait db_grp State ONLINE -sys mars01
```

Example 4. To display resource attributes that use a specified attribute as a variable, use `hagrp -infovars`. For example:

```
# hagrp -infovars g1 ContainerInfo Type
```

NOTES

The VCS One server may reject some `hagrp` commands. For example, VCS One does not allow you to bring a failover service group online on a system if the group is online elsewhere in the VCS One cluster, or if the group is faulted on that system.

When using the command to specify or modify an attribute's value that begins with a dash ("-"), precede the value with a percent sign ("%"). For example, specify `-y` as `%-y`. Likewise, precede a value that starts with a percent sign with another percent sign.

SEE ALSO

`hares`(1M), `haclus`(1M), `haconf`(1M), `halogin`(1M), `hagtq`(1M)

hagtq

`hagtq` – manage the VCS One group transition queue

SYNOPSIS

```
UNIX: /opt/VRTSvcsone/bin/hagtq

Windows: %VCSONE_HOME%\bin\hagtq

hagtq -display [-action] [-user user@domain -domaintype domaintype]
hagtq -displayplan [-user user@domain -domaintype domaintype]
hagtq -abortaction action_name [-user user@domain -domaintype
domaintype]
hagtq -flushall [-clearistate] [-user user@domain -domaintype
domaintype]
hagtq -aborttree action_name [-user user@domain -domaintype
domaintype]
hagtq -nokickout action_name [-user user@domain -domaintype
domaintype]
hagtq -version
hagtq [-help]
```

AVAILABILITY

VRTSvcsonec

DESCRIPTION

You can use the `hagtq` command to interact with and manage the VCS One Group Transition Queue (GTQ), a structure that describes the actions planned for handling service groups affected by resource faults, system faults, and others.

When VCS One must move a service group from one system to another, it creates a GTQEntry in the GTQ. The GTQEntry lists actions required for the transition. The actions have dependencies on other actions. For example, VCS One must take a group offline from a system before it can place it online on another system. Also, if the service group has a dependent parent, VCS One must take the parent offline first. Likewise, VCS One must also take a child group offline if a faulted parent has a hard dependency on it.

VCS One creates a GTQEntry for each set of affected service groups that must fail over together from one system to another. For example, groups having local

dependencies must fail over together. When a service group has a global dependency on another group, VCS One creates two GTQEntries. When a system hosting several online service groups faults, VCS One can create several GTQEntries.

Each GTQEntry contains ActionEntries for each of the operations, such as offline and online, in the GTQ. An ActionEntry describes the type of operation, the service group, and the system.

In addition to the online and offline ActionEntries, VCS One uses the *Intentonline* ActionEntry. When users issue the `hagrp -online` *`group`* `-any` command and VCS One cannot place the group online immediately, VCS One creates a GTQEntry with the intentonline action. Because intentonline action cannot be executed, VCS One converts the entry to be an online action in the future when certain events occur, such as when a new system joins the VCS One cluster or a system's capacity is increased. If VCS One cannot find a target for the online operation, it converts the ActionEntry back to intentonline.

The `-display` and `-displayplan` options are available to show the current GTQEntries in the GTQ and to show the current actions planned.

A non-root user who has not run the `halogin` command can execute the `hagtq` command using the `-user` *`user@domain`* option to execute the command with the privileges of the specified user. When issuing the command, the user must enter the fully qualified domain user name and supply a password when prompted. If necessary, the `-domaintype` option can specify the type of domain against which the user is to be authenticated. Supported domain types include:

- "unixpwd"
- "nt"
- "nis"
- "nisplus"
- "ldap"
- "pam"
- "vx" (Symantec Private Domain)

The domain type, by default, is "vx". The domain type is case sensitive.

When using `domaintype=unixpwd`, provide the system name as the domain portion. The domain must be a fully-qualified domain name (for example, sun01.engba.veritas.com).

OPTIONS

`-display [-action]`

Display the current GTQEntry information for all groups in transition. Using the `-action` option displays ActionEntry information for all ActionEntries.

`-displayplan`

Display the current actions planned in the GTQ. The display shows the planned sequence of actions listed in the GTQ. The listed actions show the dependencies among the actions.

Example output may resemble:

```
Action4[g2 offline n2] -> Action2[g4 offline n2]-> Action1[g1
online n2]

Action5[g3 offline n2] -> Action2[g4 offline n2] -> Action1[g1
online n2]

Action6[g0 offline n2] -> Action1[g1 online n2]

Action7[g4 offline n1] -> Action3[g1 offline n1]-> Action1[g1
online n2]
```

`-abortaction` *`action_name`*

Aborts or removes the specified action and its dependent actions. The command aborts actions already started and removes actions not yet started from the GTQ action plan.

`-flushall [-clearistate]`

Aborts or removes all actions in the GTQ action plan. The command aborts actions already started and removes actions not yet started from the GTQ action plan.

`-aborttree` *`action_name`*

Aborts or removes the specified action and all actions in the action dependency path.

`-nokickout` *`action_name`*

Update the GTQ plan such that any online action does not depend on an offline action or on any of its dependent offline actions. For future online actions, none can depend on any specified offline action and its dependent offline actions.

`-version`

Display the version of the command.

`[-help]`

Display usage for the `hagtq` command.

EXAMPLES

To display the usage syntax for a specific command option, enter the command and an option without arguments. For example, to see the usage syntax for `-abortaction` enter:

```
# hagtq -abortaction
```

SEE ALSO

`hagrp`(1M), `halogin`(1M)

haldapconf

haldapconf – a CLI program that facilitates configuring the LDAP plug-in for the authentication broker in VCS One

SYNOPSIS

```
UNIX: /opt/VRTSvcsone/bin/haldapconf

Windows: %VCSONE_HOME%\bin\haldapconf

haldapconf -d -s ldap_server_name [-p ldap_server_port] -u search_user
-g search_group [-f attribute_list_file] [-m admin_username] [-w
admin_password] [-l loglevel]
haldapconf -c -d domain_name [-i attribute_list_file] [-o at_cli_file]
[-a FLAT|BOB] [-s BASE|ONE|SUB] [-l loglevel]
haldapconf -x [-f at_cli_file] [-p at_install_path] [-o broker_port]
[-l loglevel]
haldapconf -h
```

AVAILABILITY

VRTSvcsone

DESCRIPTION

The LDAP configuration tool, haldapconf, is a command line interface (CLI) program that facilitates configuring the LDAP plug-in for the authentication broker. Use haldapconf to connect to the enterprise LDAP server and detect the default parameters for searching users and groups.

The haldapconf configuration tool has the following options:

- -d stands for "discover."
- -c stands for "createcli" or create an authentication CLI. The authentication CLI is used to register the LDAP server in the VCS One authentication broker.
- -x stands for "atconfigure" or configure authentication.

To configure the LDAP plug-in for the authentication broker, use these command options in the following order:

Step 1: Run haldapconf with -d. The -d command option connects to the LDAP server and searches for values of attributes that the server supports. The command

verifies if the attributes exist on the server by comparing them with values from a pre-defined list.

The `-d` command retrieves an LDAP properties file that contains a prioritized attribute list. The command parses the attribute list, selects the attribute with the highest priority, and creates a CLI that includes the `haat addldapdomain` command.

Step 2: Use the `haldapconf -c` command to edit the order of priority in the prioritized attribute list created in Step 1 and create a CLI that includes the `haat addldapdomain` command.

Step 3: Use the `haldapconf -x` command to read the AT CLI file generated in Step 2 and execute it to add an LDAP authentication domain.

OPTIONS

`-d -s` *`ldap_server_name`* `[-p` *`ldap_server_port`*`] -u` *`search_user`* `-g` *`search_group`*`[-f` *`attribute_list_file`*`] [-m` *`admin_username`*`] [-w` *`admin_password`*`] [-l` *`loglevel`*`]`

Use the `-d` command, which stands for "discover," to connect to the LDAP server. This command searches the attributes of the user and the group. It creates an attribute list file that contains the valid values for all the attributes in a descending order of priority. You can change the order of priority.

The `-d` command also retrieves the valid values for the LDAP attributes that have multiple values, such as ObjectClass. Other attributes of the LDAP directory are configurable.

You can also search the commonly used attributes that exist on the server and put all the valid attributes in an attributes list file. The commonly used attributes differ for different LDAP implementations. These values are pre-defined in separate lists for each LDAP implementation. The pre-defined values are defined in a header file. For example, the list for user gid attributes looks similar to the following:

{"gidNumber", "memberOf", "gid", ""}

`-s` *`ldap_server_name`*

Specifies the name of the LDAP server. This option is required.

`-p` *`ldap_server_port`*

Specifies the port of the LDAP server. The default value is 389. To bind to the server, the command uses the user name and password. If you do not provide a user name and password, the command prompts you to provide them.

`-u search_user`

Specifies the base search paths for users. This option is required.

`-g search_group`

Specifies the base search paths for groups. This option is required.

`-f attribute_list_file`

Specifies the name of the attribute list file. By default, the name is AttributeList.txt. This file is placed in the working directory.

`-m admin_username`

Specifies the user name of the connecting user. This option is required to make the initial connection to the LDAP server when the anonymous searches are disabled.

`-w admin_passwd`

Specifies the password of the connecting user. This option is required to make the initial connection to the LDAP server when anonymous searches are disabled.

`-l loglevel`

Generates a log file named haldapconf.debug. The log level determines the amount of information that goes into the log. The value of *loglevel* is a number between 0 and 4. 0 indicates no logging and 4 indicates the highest level of logging.

For example, to run `haldapconf -d` for an LDAP server named ldapserver.com, a user named testuser, and a group named testgroup, you would enter:

```
/opt/VRTSvcsone/bin/haldapconf -d -s ldapserver.com –u testuser –g
testgroup
```

`-c -d domain_name [-i attribute_list_file] [-o at_cli_file] [-a FLAT|BOB] [-s BASE|ONE|SUB] [-lloglevel]`

Use this command to take the attribute list generated by the discover command as input. The command parses the attribute list file and selects the attribute with the highest priority and creates a CLI file complete with haat `addldapdomain`.

`-d domain_name`

Specifies the domain name. The domain name must be unique.

`-i attribute_list_file`

Specifies the name of the attribute list file. By default, the name is AttributeList.txt. This file is placed in the working directory.

`-o at_cli_file`

Specifies the name of the AT CLI file. By default, the name is CLI.txt. This file is placed in the working directory.

`-a FLAT|BOB`

Specifies the type of authentication. FLAT specifies that the database structure for LDAP is flat or non-hierarchical. BOB specifies that the database structure for LDAP is nested or hierarchical. By default, the authentication type is FLAT.

`-s BASE|ONE|SUB`

Specifies the scope of the search. BASE is the primary level, ONE is one down from the primary level, and SUB is below ONE. By default, the scope is SUB.

`-l log_level`

Generates a log file named haldapconf.debug. The log level determines the amount of information that goes into the log. The value of *log_level* ranges from 0 to 4. 0 indicates no logging and 4 indicates the highest level of logging.

For example, to run `haldapconf -c` for a domain named myldapdomain1, you would enter:

```
/opt/VRTSvcsone/bin/haldapconf -c -d myldapdomain1
```

`-x [-f at_cli_file] [-p at_install_path] [-o broker_port] [-l loglevel] ]`

Use this command to read and execute the AT CLI that was generated by the `haldap -c` command and add the domain to AT.

`-f at_cli_file`

Specifies the name of the AT CLI file. By default, the file name is CLI.txt. This file is placed in the working directory.

`-p at_install_path`

Specifies the path where AT is installed. For VCS One, the path is /opt/VRTSvcsone.

`-o broker_port`

Specifies the broker port. By default for VCS One, the broker port is 14159, unless you specifically change the broker port when you install VCS One.

`-l log_level`

Generates a log file named haldapconf.debug. The log level determines the amount of information that goes into the log. The value of *log_level* ranges from 0 to 4. 0 indicates no logging and 4 indicates the highest level of logging.

For example, to run `haldapconf -x` for the default broker port for VCS One, you would enter:

```
/opt/VRTSvcsone/bin/haldapconf -x -o 14159 -p /opt/VRTSvcsone
```

`-h` Displays usage for the `haldapconf` command.

SEE ALSO

`haat`(1M)

halog

`halog` – add messages to the VCS One engine log

SYNOPSIS

```
UNIX: /opt/VRTSvcsone/bin/halog

Windows: %VCSONE_HOME%\bin\halog

halog -add message -sev C | E | W | N | I [-sys system] [-msgid
messageid [-parameters parameter(s)]] [-encoding encoding] [-user
user@domain -domaintype domaintype]
halog -add message -dbg 1-21 [-sys system] [-msgid messageid
[-encoding encoding] [-parameters parameter(s)]] [-user user@domain
-domaintype domaintype]
halog -version
halog [-help]
```

AVAILABILITY

VRTSvcsonec

DESCRIPTION

The `halog` command adds messages to the engine log. The `halog` command is also used internally by agent entry points to log messages written in Perl or Shell script.

The `-addtags`, `-deltags`, and `-info` options are no longer supported. These command options will still work for a period of time so that any pre-existing customer scripts that use them will not break.

A non-root user who has not run the `halogin` command can execute the `halog` command using the `-user` *`user@domain`* option to execute the command with the privileges of the specified user. When issuing the command, the user must enter the fully qualified domain user name and supply a password when prompted. If necessary, the `-domaintype` option can specify the type of domain against which the user is to be authenticated. Supported domain types include:

- "unixpwd"
- "nt"
- "nis"

- "nisplus"
- "ldap"
- "pam"
- "vx" (Symantec Private Domain)

The domain type, by default, is "vx". The domain type is case sensitive.

When using `domaintype=unixpwd`, provide the system name as the domain portion. The domain must be a fully-qualified domain name (for example, sun01.engba.veritas.com).

OPTIONS

`-add` *`message`* `-sev C | E | W | N | I [-sys` *`system`*`] [-msgid` *`messageid`* `[-parameters` *`parameter(s)`*`]] [-encoding` *`encoding`*`] [-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Add a message of a specified severity from the command line to the engine log.

The severity values have the following significance: C = Critical, E = Error, W = Warning, N = Notice, and I = Information.

`-sys` specifies a system.

`-msgid` is the message number.

`-encoding` is an encoding format supported by the platform.

`-parameters` specify parameter arguments. Parameters must not exceed 4096 bytes. If the total exceeds 4096 bytes, then each argument is allowed an equal portion of 4096 bytes and is truncated if it exceeds the allowed portion.

`-add` *`message`* `-dbg` *`1-21`* `[-sys` *`system`*`] [-msgid` *`messageid`* `[-encoding` *`encoding`*`] [-parameters` *`parameter(s)`*`]] [-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Add debug log information at levels 1 to 21 from the command line to the log file.

`-sys` specifies a system.

`-msgid` is the message number.

`-encoding` is an encoding format supported by the platform.

`-parameters` specify parameter arguments. Parameters must not exceed 4096 bytes. If the total exceeds 4096 bytes, then each argument is allowed an equal portion of 4096 bytes and is truncated if it exceeds the allowed portion.

`-version`

Display the version of the command.

`[-help]`

Display usage for the `halog` command. When you enter the command and an option without arguments, syntax for the specific option displays.

EXAMPLES

Add a debug message and show that it is enabled.

```
% halog -add DBG_TRACE
```

Add a message of a specified severity to the engine log.

```
% halog -add "This is an application message" -sev N
```

Add a debug message of a specified level.

```
% halog -add "This is a debug message" -dbg 2
```

Add a debug message, specify its message number, and a parameter argument.

```
% halog -add "This is an application message for group1" \
-msgid 11057 -parameters group1
```

Obtain the usage for a command option by entering the command and the option without arguments.

```
% halog -add
```

SEE ALSO

`hares`(1M), `halogin`(1M)

halogin

`halogin` – enables users to authenticate themselves in VCS One environments for the purpose of executing VCS One commands

SYNOPSIS

```
UNIX: /opt/VRTSvcsone/bin/halogin

Windows: %VCSONE_HOME%\bin\halogin

halogin [-forclient][-passwd password] -user user@domain -domaintype
domaintype
halogin -endsession PM IP
halogin -endallsessions
halogin -version
halogin -help
```

AVAILABILITY

VRTSvcsonec

DESCRIPTION

The execution of VCS One commands requires secure communications between a VCS One client system and the VCS One Policy Master.

The `halogin` command line utility provides a user the means, via Symantec Product Authentication Service (AT), to obtain a valid credential and to be authenticated in a secure VCS One cluster environment. A user enters the `halogin` command and provides a password, a fully qualified user name, and a domain type. When the user is authenticated, the user credential is cached on the disk and the utility creates a profile (in the file `.vcsoneprofile`) in the user's home directory. The credential and the stored profile provide the means to validate the commands issued by the user. User credentials last twenty four hours, typically.

The commands you issue must be permitted by the roles assigned to you by the administrator (either in the GUI or by using the `hauser` command).

If users do not use `halogin` to set up a valid user profile, they may authenticate themselves by defining the VCSONE_USERNAME and VCSONE_DOMAINTYPE environment variables. A password is still required to enter commands.

Other environment variables that may be required are VCSONE_SERVER_IP, which can be used to specify the Policy Master IP address if it is different from

the IP addresses specified in `.conf`, and VCSONE_BROKER_HOST, which can be used to specify the Authentication Broker IP address if it is different from the Policy Master IP address.

If users do not use `halogin` to set up a valid user profile, and do not set their VCSONE_USERNAME and VCSONE_DOMAINTYPE environment variables, they must enter the `-user` and `-domaintype` options when using each VCS One command. Otherwise, they are assumed to be the logged-in user and may not be privileged to use VCS One commands. The root user on the VCS One client system (localhost root user) is an exception and has the user privileges associated with the VCS One client daemon on that node. For the root user, VCS One commands ignore the profile created by `halogin` on an active Policy Master node.

Valid domain types are:

- "unixpwd"
- "nt"
- "nis"
- "nisplus"
- "ldap"
- "pam"
- "vx" (Symantec Private Domain)

The domain type, by default, is "vx". The domain type is case sensitive.

When using `domaintype=unixpwd`, provide the system name as the domain portion. The domain must be a fully-qualified domain name (for example, sun01.engba.veritas.com).

When the credential is no longer valid, the Policy Master session terminates. You can use the `-endsession` option to terminate the session.

OPTIONS

`[-forclient] [-passwd` *`password`*`] -user` *`user@domain`* `-domaintype` *`domaintype`*

Authenticate with the Policy Master as *user@domain* of the specified *domaintype* with the password *password*. Use the `-forclient` option to ensure that the user profile will be used when "ha" commands are executed within script-based entry points inside local zones so that they can connect to the Policy Master via the VCS One client daemon (vcsoneclientd). If you do not supply a password, `halogin` will prompt for it interactively.

`-endsession` *`PM IP`*

Delete the `halogin` "profile" (session credential) for the specified Policy Master host IP address (*PM IP*). The `-endsession` option searches for the Policy Master host IP address in the `~/.vcsoneprofile` file and then deletes the corresponding entry for the file.

`-endallsessions`

Delete `halogin` sessions (session credentials) for all hosts and delete the file `.vcsoneprofile`.

`-version`

Display the current version for `halogin`.

`-help`

Display options available for `halogin`.

FILES

The file `vcsone.conf` is created during installation of VCS One client daemon software on each VCS One client system. It contains information, including the Policy Master cluster virtual IP address, that enables communications with the Policy Master.

The file `~/.vcsoneprofile`, which is created and stored in the user's home directory, contains the authenticated user's profile. The profile includes the user's identity and privilege details along with the IP addresses of the Policy Master and of the authentication broker. When a user issues a command, the command uses this file to validate the requested action. The user's details are deleted from this file when the `-endsession` option is used. The file is deleted when the `-endallsessions` option is used.

EXAMPLES

In this example, the user `fred` has a UNIX/Linux account on the Policy Master system (for example, pm.domain.com) and on a client node (for example, c1.domain2.com). The cluster administrator creates the user `fred@pm.domain.com` and assigns a role to that user. The user `fred` can now log on to either the Policy Master or the client and authenticate himself using this command:

```
# halogin -user fred@pm.domain.com -passwd unix_password_for_fred
-domaintype domaintype unixpwd
```

When Fred wants to end his session, he needs to pass the host IP address of the Policy Master using the `-endsession` command option.

SEE ALSO

`hauser`(1M)

hamultisim

`hamultisim` – create and manage multiple Simulator instances

SYNOPSIS

```
hamultisim -addsim instance_name
hamultisim -removesim instance_name
hamultisim -startsim instance_name [-d xml_dir] [-dbport port]
[-pmport port] [-proxysimport port] [-sslport port] [-adminport port]
[-wssslport wsssl_port] [-extended [-no_operation]]
hamultisim -stopsim instance_name
hamultisim -cliprompt instance_name
hamultisim -list [ports]
hamultisim -status [-processes] [instance_name]
hamultisim [-help]
```

AVAILABILITY

vcsonesim

DESCRIPTION

The VCS One Simulator is available for Windows. You can install the VCS One Simulator on one or more Windows systems.

VCS One includes a single default Simulator instance. You can start any number of Simulator instances.

The `hamultisim` command controls multiple Simulator instances. You can add, remove, start, and stop Simulator instances using this command. You can also start the Windows command prompt for a specific Simulator instance, list instances, and view their status.

Each Simulator instance should use different ports. The ports a Simulator instance uses should not be used by any other process. A Simulator instance uses certain ports by default. You can specify alternate ports for a Simulator instance when you start the Simulator instance.

OPTIONS

`-addsim` *instance_name*

Adds the specified Simulator instance. Before you can start a new Simulator instance, you must add it.

https://127.0.0.1:*ssl_port*/*instance_name*

If you are running multiple Simulator instances simultaneously, use this type of URL to easily distinguish each Simulator instance's GUI. For the default VCS One cluster, access the GUI using https://127.0.0.1:*ssl_port*.

`-removesim` *instance_name*

Removes the specified Simulator instance.

`-startsim` *instance-name* `[-d` *xml_dir*`] [-dbport` *port*`] [-pmport` *port*`] [-proxysimport` *port*`] [-sslport` *port*`] [-adminport` *port*`] [-wssslport` *wsssl_port*`] [-extended -no_operation]]`

Starts a Simulator instance. Before you start the Simulator instance, make sure that you add it using the `-addsim` option.

`[-d` *xml_dir*`]`

Loads the XML configuration into the database and starts it. The Simulator includes sample configurations. They are in the following directory:

installed_location\VCSOne\Simulator\conf

`[dbport` *port*`]`

Starts the database on the port provided. If you do not specify a port, the database starts on port 14157 by default. If ths port is not available, it starts on the next available port.

`[-pmport` *port*`]`

Starts the Policy Master on the port provided. If you do not specify a port, the Policy Master starts on port 14151 by default. If this port is not avilable, it starts on the next available port.

`[-proxysimport` *port*`]`

Starts the proxysimport on the port provided. If you do not specify a port, the proxysimport starts on port 14156 by default. If this port is not available, it starts on the next available port.

`[-sslport` *port*`]`

Starts the Web server on the SSL port provided. If you do not specify a port, the Web server starts on port 14171 by default. If this port is not available, it starts on the next available port.

`[-adminport port]`

Starts the Web server on the admin port provided. If you do not specify a port, the Web server stars on port 14172 be default. If this port is not available, it starts on the next available port.

`[-wsslport port]`

Starts the Web server on the SSL port provided. If you do not specify a port, the Web server starts on port 14173 by default. If this port is not available, it starts on the next available port.

`[-extended]`

Starts the Simulator and retains the states of objects as defined in the specified database configuration. (The Simulator does not move configured systems to a RUNNING state.)

The Simulator completes commands that involve groups or resources that have an outstanding intended online state (such as INTENT_ONLINE or WAITING_ FOR_ONLINE).

`[-extended [-no_operation]]`

The `-no_operation` option starts the Simulator in read-only mode and you cannot perform write operations.

Starting the Simulator in read-only mode is useful for debugging. The systems, resources, and groups' states/istates are preserved. You can see the exact state/istate information for all the objects in the database.

`-stopsim instance_name`

Stops the specified Simulator instance and all its processes.

`-cliprompt instance_name`

Starts the command prompt for the specified Simulator instance. The commands that you run from this command prompt apply to the specified Simulator instance only.

`-list [-ports]`

Lists the Simulator instances configured in the installed location.

Use the `-ports` option with the `-list` option to list the port information for each process. It lists the instances, processes, and ports on which the process is configured.

`-status [-processes] [instance_name]`

Provides the status of the specified Simulator instance. If you do not specify an instance name, the status is displayed for all Simulator instances. An instance has one of the following statuses:

RUNNING: All the processes for the specified instance are up and the instance is running.

NOT RUNNING: All the processes for the specified instance are down and the instance is not running.

PARTIAL: Some of the processes for the specified instance are up and the instance is in a PARTIAL state.

The `-processes` option displays the status of each process for the specified instance. If you do not specify an instance, the `-processes` option displays the status of all the processes for all instances.

The process status can be one the following states:

UP: The process for the instance is running.

DOWN: The process for the instance is not running.

`[-help]`

Displays usage for the `hamultisim` command.

SEE ALSO

`hasim`(1M)

haou

haou – create and maintain the Organization Tree

SYNOPSIS

UNIX: /opt/VRTSvcsone/bin/haou

Windows: %VCSONE_HOME%\bin\haou

haou -add *ouname ouvaluepath* [-user *user@domain* -domaintype *domaintype*]
haou -delete [-force] *ounamepath* [-user *user@domain* -domaintype *domaintype*]
haou -addvalue *ouvalue(s) ounamepath* [-user *user@domain* -domaintype *domaintype*]
haou -deletevalue [-force] *ouvaluepath* [-user *user@domain* -domaintype *domaintype*]
haou -list [-tree] [*ounamepath* | *ouvaluepath*] [-user *user@domain* -domaintype *domaintype*]
haou -displayval *ounamepath(s)* [-user *user@domain* -domaintype *domaintype*]
haou -displayobj [-exclusive] [-grp] [-sys] [-userobject] [-usergroup] [-csg] [-vobject] [-pframe] [-vframe]*ouvaluepath* [-user *user@domain* -domaintype *domaintype*]
haou -version
haou -help

AVAILABILITY

VRTSvcsonec

DESCRIPTION

The haou command is used to create and maintain the Organization Tree. Use the command to add and delete *ouname* nodes, and add and delete *ouvalues* to and from the list of valid values of *ouname* nodes. You can also use the command to display the Organization Tree hierarchy, as well as list the valid values for organization unit names specified by *ouname* and the objects associated with the organization unit specified by *ouvaluepath*.

Valid domain types are:

- "unixpwd"
- "nt"
- "nis"
- "nisplus"
- "ldap"
- "pam"
- "vx" (Symantec Private Domain)

The domain type, by default, is "vx". The domain type is case sensitive.

When using `domaintype=unixpwd`, provide the system name as the domain portion. The domain must be a fully-qualified domain name (for example, sun01.engba.veritas.com).

OPTIONS

`-add` *`ouname ouvaluepath`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Adds a node specified by *ouname* to the Organization Tree under the *ouvalue* specified by *ouvaluepath*. *ouname* is the name of the node to be added in the Organization Tree. *ouvaluepath* is the location in the Organization Tree to add the node. *ouvaluepath* is denoted by a list of OUName=OUValue pairs, separated by a forward slash (/).

`-delete [-force]` *`ounamepath`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Deletes the node specified by *ounamepath*. If the node you are attempting to delete is not a leaf node, the command will not execute successfully unless the `-force` option is used. The `-force` option causes the entire subtree to be deleted.

`-addvalue` *`ouvalue(s) ounamepath`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Adds *ouvalue* to the list of valid values for the *ouname* specified by *ounamepath*. *ouvalue* is the value of the *ouname* node above it in the Organization Tree. *ounamepath* is the location in the Organization Tree to add the value, as denoted by an Organization Tree path that ends in an OUName. The Organization Tree path is denoted by a list of OUName=OUValue pairs, separated by a forward slash (/).

`-deletevalue [-force]` *`ouvaluepath`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Deletes the *ouvalue* node specified by *ouvaluepath*. If the deleted node is not a leaf node, the command will be rejected unless the `-force` option is specified.

The `-force` option causes the entire subtree to be deleted. All objects associated with the *ouvalue* will be moved to the parent *ouvalue* (that is, the parent of the parent *ouname*).

`-list [-tree] [ounamepath | ouvaluepath] [-user user@domain -domaintype domaintype]`

Displays the Organization Tree hierarchy from the *ouname* or *ouvalue* specified by *ounamepath* or *ouvaluepath*. Use the `-tree` option to display the output in "tree" format.

`-displayval ounamepath(s) [-user user@domain -domaintype domaintype]`

Displays the list of valid values for the *ouname* specified by *ounamepath*. You may specify multiple *ounamepaths* with `-displayval`.

`-displayobj [-exclusive] [-grp] [-sys] [-userobject] [-usergroup] [-csg] [-vobject] [-pframe] [-vframe] ouvaluepath [-user user@domain -domaintype domaintype]`

Displays the objects associated with the organization unit corresponding to *ouvaluepath*. If the `-exclusive` option is not specified, the command will display all the objects in the subtree. If the `-exclusive` option is specified, the command will display only those objects at that *ouvaluepath*.

`-version`

Displays version information for the command.

`[-help]`

Displays usage for the `haou` command.

EXAMPLES

To create a new line of business (lob) and associate a value with it, enter:

```
# haou -add lob /

# haou -addvalue dcmb /lob
```

To list the organization units that have been defined, enter:

```
# haou -list

/lob

/lob=dcmg

/lob=dcmg/dept

/lob=dcmg/dept=vcs

/lob=dcmg/dept=vcsone
```

```
/lob=consumer
```

To display the Organizational Tree structure, enter:

```
# haou -list -tree

/lob

|----dcmg

| |----dept

| | |----vcs

| | |----vcsone

|----consumer
```

To display defined OUValues, enter:

```
# haou -displayobj /

OUValue: /

-----------

Groups:

Test_Group1

Test_Mount2

Systems:

Test_System1

Test_System2

OUValue: /lob=dcmg

-------------------

Groups:

g1

g2

OUValue: /lob=dcmg/dept=vcs

----------------------------

Groups: g3

Usergroups:

u2@d1
```

```
OUValue: /lob=dcmg/dept=vcsone

---------------------------

Groups:

g4

Users:

u1@d1
```

To display values for specific organization units, enter:

```
# haou -displayval /lob /lob=dcmg/dept

/lob

/lob=dcmg

/lob=consumer

/lob=dcmg/dept

/lob=dcmg/dept=vcs

/lob=dcmg/dept=vcsone
```

To delete an organization unit value, enter:

```
# haou -deletevalue /lob=consumer
```

To delete an organization unit name by force, enter:

```
# haou -delete -force /lob=dcmg/dept
```

NOTES

When using the command to specify or modify an attribute's value that begins with a dash ("-"), precede the value with a percent sign ("%"). For example, specify `-y` as `%-y`. Likewise, precede a value that starts with a percent sign with another percent sign.

SEE ALSO

`haea`(1M), `haset`(1M)

hares

`hares` – manage individual resources that make up service groups in the VCS One cluster

SYNOPSIS

UNIX: /opt/VRTSvcsone/bin/hares

Windows: %VCSONE_HOME%\bin\hares

hares -add *resource type group* [-user *user@domain* -domaintype *domaintype*]
hares -delete *resource* [-user *user@domain* -domaintype *domaintype*]
hares -local *resource attribute* [-user *user@domain* -domaintype *domaintype*]
hares -global *resource attribute* [-user *user@domain* -domaintype *domaintype*]
hares -action *resource token* [-actionargs *arg1 arg2*...] -sys *system* [-user *user@domain* -domaintype *domaintype*]
hares -link *parentresource childresource* [-user *user@domain* -domaintype *domaintype*]
hares -unlink *parentresource childresource* [-user *user@domain* -domaintype *domaintype*]
hares -dep [*resource(s)*] [-user *user@domain* -domaintype *domaintype*]
hares -clear *resource* [-sys *system*] [-user *user@domain* -domaintype *domaintype*]
hares -clearadminwait [-fault] *resource* -sys *system* [-user *user@domain* -domaintype *domaintype*]
hares -refreshinfo *resource* -sys *system* [-user *user@domain* -domaintype *domaintype*]
hares -flushinfo *resource* [-sys *system*] [-user *user@domain* -domaintype *domaintype*]
hares -probe *resource* -sys *system* [-user *user@domain* -domaintype *domaintype*]
hares -online *resource* -sys *system* [-user *user@domain* -domaintype *domaintype*]
hares -offline [-propagate] [-ignoreparent] *resource* -sys *system* [-user *user@domain* -domaintype *domaintype*]
hares -override *resource staticattribute* [-user *user@domain* -domaintype *domaintype*]

```
hares -undo_override resource staticattribute [-user user@domain
-domaintype domaintype]
hares -display [resource(s)] [-attribute attribute(s)] [-grp group(s)]
[-type type(s)] [-sys {systems | -ou ouexpression | -ea eaexpression
| -ou ouexpression -ea eaexpression | -setname setname}] [-user
user@domain -domaintype domaintype]
hares -display -ovalues [resource(s)] [-grp {group(s) | -ou
ouexpression | -ea expression | -ou ouexpression -ea expression |
-setname setname}] [-type type(s)] [-platform platform(s)] [-user
user@domain -domaintype domaintype]
hares -list [conditional(s)] [-user user@domain -domaintype
domaintype]
hares -state [resource(s)] [-sys {system(s) | -ou ouexpression | -ea
eaexpression | -ou ouexpression -ea eaexpression | -setname setname}]
[-user user@domain -domaintype domaintype]
hares -value resource attribute [-sys system] [-user user@domain
-domaintype domaintype]
hares -verifyvars resource attribute [-user user@domain -domaintype
domaintype]
hares -wait resource attribute value [-sys system] [-time seconds]
[-user user@domain -domaintype domaintype]
hares modify_options
hares [-help [ -modify | -list]]
hares -version
```

AVAILABILITY

VRTSvcsonec

DESCRIPTION

The `hares` command administers resources in the VCS One cluster. Resources are individual representations of the elements required for a service group to be available, such as a volume, a database, or an IP address.

For the `-platform` option, supported values for *platform* are:

- aix
- aix/rs6000 (alias aix)
- esx
- hpux

- linux
- linux/x86 (alias linux)
- solaris
- solaris/x86
- solaris/sparc (alias solaris)
- windows
- windows/x86

Use the explicit platform name when no alias is defined. When *platform* appears in any displays, the full name and not the alias is shown.

A non-root user who has not run the `halogin` command can execute the `hares` command using the `-user` *`user@domain`* option to execute the command with the privileges of the specified user. When issuing the command, the user must enter the fully qualified domain user name and supply a password when prompted. If necessary, the `-domaintype` option can specify the type of domain against which the user is to be authenticated. Supported domain types include:

- "unixpwd"
- "nt"
- "nis"
- "nisplus"
- "ldap"
- "pam"
- "vx" (Symantec Private Domain)

The domain type, by default, is "vx". The domain type is case sensitive.

See NOTES for how to specify "-" and "%" characters in the command line.

OPTIONS

`-add` *`resource type group`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Add a resource (*resource*) of type (*type*), which is a member of the group specified by *group*.

`-delete` *`resource`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Delete a resource from the configuration. The resource must be `offline`.

`-local` *`resource attribute`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Localize an attribute. That is, the current value is converted to an association in which the keys are the systems of the resource group's `SystemList` attribute. Localized attributes may have a different value for each system in the `SystemList`.

`-global` *`resource attribute`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Change the scope of a local attribute (one that has a value or set of values for every system on which a resource's group is configured to run) to the scope of a global attribute (a single value or set of values for all systems).

`-action` *`resource token`* `[-actionargs` *`arg(s)`*`] -sys` *`system`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Specifies that an action corresponding to the *token* be taken by the agent for the specified resource. A system is required.

token is one of a set of customized actions indicated in the resource type definition. Agent developers are responsible for defining the actions and initializing the static attribute SupportedActions in the resource type definition. If arguments are required for the indicated action, they may be specified using the optional `-actionargs` flag. See the documentation provided with the agent for information about arguments for specific actions.

`-link` *`parentresource childresource`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Specify a dependency between two resources. The parent resource depends on the child; that is, the child is brought online before the parent resource, but the parent resource is taken offline before the child.

`-unlink` *`parentresource childresource`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Remove the dependency between two resources.

`-dep [`*`resource(s)`*`] [-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Displays dependency information about the specified resource(s). If *resource(s)* is omitted, dependency information for all resources is displayed.

`-clear` *`resource`* `[-sys` *`system`*`] [-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Clear a resource fault by changing the state from `faulted` to `offline`. If no system is specified, the resource is cleared on all systems on which it is faulted. This command automatically clears all faulted resources that depend directly or indirectly (that is, resources that have parents in the dependency tree) on the specified resource.

`-clearadminwait [-fault] resource -sys system [-user user@domain -domaintype domaintype]`

Clears the `ADMIN_WAIT` state of the specified resource on the specified system. If the resource continues in the `ADMIN_WAIT` state, use the `-fault` option to clear the state. The command sets the state to `ONLINE | UNABLE_TO_OFFLINE` or `FAULTED`, depending on the reasons the `ResAdminWait` trigger had been called.

Note that the `online`, `offline`, `switch`, and `flush` operations cannot be performed on resources in the `ADMIN_WAIT` state. Also, when resources are in the `ADMIN_WAIT` state, the `hastop` command requires the `-force` option.

`-refreshinfo resource -sys system [-user user@domain -domaintype domaintype]`

The `-refreshinfo` option causes the Info entrypoint to update the value of the ResourceInfo resource level attribute for the specified resource if the resource is online. If the Info entrypoint is successful, no output is displayed. If the Info entrypoint fails, the output of `-refreshinfo` contains the text of the returned error. The Info entrypoint runs only if the resource is online on the system; if the resource is not online on the specified system, the refreshinfo command fails.

`-flushinfo resource [-sys system] [-user user@domain -domaintype domaintype]`

Causes the clearing of current values of the ResourceInfo resource level attribute for the specified resource. The resource need not be online to run this command. The default value for the ResourceInfo attribute, which is restored as a result of running this command, is represented by three string-association keys: State=valid, Msg="", TS="*current_date_and_time*". If the ResourceInfo attribute is global, a system need not be specified; the attribute is reset for the resource on all systems in the VCS One cluster. If the ResourceInfo attribute is local, the system for which the ResourceInfo attribute should be flushed must be specified, and its value is reset only for the specified system.

`-probe resource -sys system [-user user@domain -domaintype domaintype]`

Monitor the resource on the specified system. The VCS One client daemon sends the state of the resource to the VCS One Policy Master, which takes the appropriate action.

`-online resource -sys system [-user user@domain -domaintype domaintype]`

Bring a resource online on the specified system. All child resources are first brought online, if they are not already online.

`-offline [-propagate] [-ignoreparent] resource -sys system [-user user@domain -domaintype domaintype]`

Take a resource offline on the specified system. Use the `-propagate` option to take a parent resource and child resources offline concurrently on the specified system. The `-ignoreparent` option allows the parent resources to remain online.

`-override resource staticattribute [-user user@domain -domaintype domaintype]`

For a given resource, permit a static resource type attribute to be overridden. After using this command, use the `modify` option to modify the value. You can use the `display` option to see values of overridden attributes. The override attribute can be removed using the `-undo_override` option.

`-undo_override resource staticattribute [-user user@domain -domaintype domaintype]`

Remove the overridden static attribute from the resource's list of attributes.

`-display [resource(s)] [-attribute attribute(s)] [-grp group(s)] [-type type(s)] [-sys {systems | -ou ouexpression | -ea eaexpression | -ou ouexpression -ea eaexpression | -setname setname}] [-user user@domain -domaintype domaintype]`

Display resource attribute values for the specified *resource(s)*, *group(s)*, *type(s)*, *system(s)*, *attribute(s)*, or *ouexpression* and/or *eaexpression*. Multiple options may be used. If no option is specified, attribute values for all resources are displayed, including overridden values.

Arguments for the `-ou` and `-ea` command options must be enclosed in double quotes if they contain spaces. For example:

```
hares -display -ou "/lob=DCMG /lob=VCS" -attribute SystemList
```

An extended attribute value cannot contain a comma.

In addition, an extended attribute value or validation set cannot contain a single quote (') character. The single quote character serves as a delimiter for the value in an EA expression. However, single quotes can be used to specify a multiword extended attribute value in an EA expression. For example:

```
hares -display -ea "ea1= 'new value' and ea2= 'new value2'"
```

`-display -ovalues [resource(s)] [-grp {group(s) | -ou ouexpression | -ea expression | -ou ouexpression -ea expression | -setname setname}] [-type type(s)] [-platform platform(s)] [-user user@domain -domaintype domaintype]`

Display overridden resource attribute values for the specified *resource(s)*, *group(s)*, *type(s)*, *system(s)*, *attribute(s)*, or *ouexpression* and/or *eaexpression*. Multiple options may be used. If no option is specified, overridden values for all resources are displayed.

`-list [conditional(s)] [-user user@domain -domaintype domaintype]`

Displays a list of resources whose values match given conditional Attribute=Value, Attribute!=Value, Attribute=~Value. Multiple conditional statements imply AND logic. If no conditional statement is specified, all resources in the VCS One cluster are listed.

`-state [resource(s)] [-sys {system(s) | -ou ouexpression | -ea eaexpression | -ou ouexpression -ea eaexpression | -setname setname}] [-user user@domain -domaintype domaintype]`

Return the current state of the specified resource for the specified system, OU expression (*ouexpression*) and/or EA expression (*eaexpression*), or set.

Arguments for the `-ou` and `-ea` command options must be enclosed in double quotes if they contain spaces. For example:

```
hares -display -ou "/lob=DCMG /lob=VCS" -attribute SystemList
```

An extended attribute value cannot contain a comma.

In addition, an extended attribute value or validation set cannot contain a single quote (') character. The single quote character serves as a delimiter for the value in an EA expression. However, single quotes can be used to enclose a multiword extended attribute value in an EA expression. For example:

```
hares -display -ea "ea1= 'new value' and ea2= 'new value2'"
```

`-value resource attribute [-sys system] [-user user@domain -domaintype domaintype]`

The `-value` option is used instead of the `-display` option when one specific attribute value is needed rather than a table of many attribute values.

For example, `hares -value File9 State sysb` displays the value of the `State` attribute for resource `File9` on system `sysb`. The system name must be specified for local attribute values but not for global attribute values.

`-verifyvars` *resource_attribute* [`-user` *user@domain* `-domaintype` *domaintype*]

When you use variables in a keylist or an association attribute, duplicate or empty keys can result. If this occurs, you can modify the variable values to fix the issue. Use `hares -verifyvars` to verify that the issue has been fixed.

`-wait` *resource attribute value* [`-sys` *system*] [`-time` *seconds*] [`-user` *user@domain* `-domaintype` *domaintype*]

The `-wait` option is for use in scripts to direct the `hares` command to wait until the value of the attribute is changed as specified, or until the time specified by *seconds* has been reached. *seconds* is an integer specifying seconds.

The `-wait` option can be used only with changes to scalar attributes. The `-sys` option can be applied only when the scope of the attribute is local.

See EXAMPLES.

`-modify` *modify_options*

The `-modify` option lets you modify a resource's attributes.

You may modify a scalar attribute's existing value.

You may also add variables as valid resource attribute values. A variable can be a system attribute, an extended attribute defined for a system, or a common extended attribute. You can use variables only when the resource attribute is a scalar and the data type is a string. Variables cannot be specified as the default value of a resource attribute.

When variables are used as resource attribute values, you do not need to implicitly specify the local attributes for the resource or manually update them every time they change.

You may not use `-modify` to change values already defined for a vector, a keylist, or an association attribute. For vector, keylist, and association attributes, use the *modify_options*, which include `-add`, `-delete`, `-update`, or `-delete -keys`. Refer to the following list of permissible `-modify` commands. You may display the commands by using `-hares -help -modify`.

SCALAR

`hares -modify` *resource attribute value* [`-sys` *system*] [`-user` *user@domain* `-domaintype` *domaintype*]

To specify a variable in the value, *value*, use @{*variable*}. For example, to add a variable to a resource attribute enter:

`hares -modify` *resource attribute* `@{`*variable*`}`

The escape character for a resource attribute variable is a caret "^" and is used before the @ sign, for example, ^@{*variable*}.

VECTOR

Use the following command only when the vector attribute has no value:

```
hares -modify resource attribute value...[-sys system] [-user
user@domain -domaintype domaintype]
```

For vector attributes that have values defined, use only the following allowed operations.

```
hares -modify resource attribute -add value... [-sys system]
[-user user@domain -domaintype domaintype]
```

```
hares -modify resource attribute -delete -keys [-sys system]
[-user user@domain -domaintype domaintype]
```

Note: You cannot delete an individual element of a VECTOR.

To specify a variable in the value, *value*, use @{*variable*}. For example, to add a variable to a resource attribute enter:

```
hares -modify resource attribute @{variable}
```

KEYLIST

Use the following command only when the keylist attribute has no value:

```
hares -modify resource attribute key... [-sys system] [-user
user@domain -domaintype domaintype]
```

For keylist attributes that have values defined, use only the following allowed operations.

```
hares -modify resource attribute -add key... [-sys system]
[-user user@domain -domaintype domaintype]
```

```
hares -modify resource attribute -delete key... [-sys system]
[-user user@domain -domaintype domaintype]
```

```
hares -modify resource attribute -delete -keys [-sys system]
[-user user@domain -domaintype domaintype]
```

To specify a variable in the value, *value*, use @{*variable*}. For example, to add a variable to a resource attribute enter:

```
hares -modify resource attribute @{variable}
```

ASSOCIATION

Use the following command only when the association attribute has no value:

```
hares -modify resource attribute {key value}... [-sys system]
[-user user@domain -domaintype domaintype]
```

For association attributes that have values defined, use only the following allowed operations.

```
hares -modify resource attribute -add {key value}... [-sys
system] [-user user@domain -domaintype domaintype]
```

```
hares -modify resource attribute -update {key value}... [-sys
system] [-user user@domain -domaintype domaintype]
```

```
hares -modify resource attribute -delete key... [-sys system]
[-user user@domain -domaintype domaintype]
```

```
hares -modify resource attribute -delete -keys [-sys system]
[-user user@domain -domaintype domaintype]
```

To specify a variable in the value, *value*, use @{*variable*}. For example, to add a variable to a resource attribute enter:

```
hares -modify resource attribute @{variable}
```

[-help [-modify | -list]]

Display usage for the hares command. When you enter the command and an option without arguments, syntax for the specific option displays.

The -modify option displays usage for the -modify option. The -list option displays usage for the -list option.

-version

Display the version of hares.

EXAMPLES

To display the usage syntax for a specific command option, enter the command and an option without arguments. For example, enter:

```
# hares -value
```

To online the resource db_volume on the system mars01, enter:

```
# hares -online db_volume -sys mars01
```

From a script, to direct the hares command to wait until the STATE attribute of the db_volume changes to the value ONLINE on system mars01, enter:

```
# hares -wait db_volume State ONLINE -sys mars01
```

NOTES

In some instances, VCS One may ignore `hares` commands. For example, VCS One does not allow you to online a resource that is part of a failover service group on a system if the group is active (at least one resource is online, or waiting to go online) elsewhere in the VCS One cluster.

A resource may be a member of only one group.

Resource names need not be unique throughout the VCS One cluster.

When using the command to specify or modify an attribute's value that begins with a dash ("-"), precede the value with a percent sign ("%"). For example, specify `-y` as `%-y`. Likewise, precede a value that starts with a percent sign with another percent sign.

SEE ALSO

`hagrp`(1M), `halogin`(1M)

harole

harole – Display information about roles, create and delete custom roles, and add or delete the privileges associated with roles

SYNOPSIS

UNIX: /opt/VRTSvcsone/bin/harole

Windows: %VCSONE_HOME%\bin\harole

harole -add *rolename* {-type *roletype* | -inherit *rolename*} [-desc *description*] [-user *user@domain* -domaintype *domaintype*]
harole -delete *rolename* [-user *user@domain* -domaintype *domaintype*]
harole -addpriv *rolename operation(s)* [-user *user@domain* -domaintype *domaintype*]
harole -delpriv *rolename operation(s)* [-user *user@domain* -domaintype *domaintype*]
harole -rollback *rolename* [-user *user@domain* -domaintype *domaintype*]
harole -display [-all | *role(s)*] [-attribute *attribute(s)*] [-user *user@domain* -domaintype *domaintype*]
harole -list [-all] [*conditional(s)*] [-user *user@domain* -domaintype *domaintype*]
harole -value *rolename attribute* [-user *user@domain* -domaintype *domaintype*]
harole -modify *rolename attribute value* [-user *user@domain* -domaintype *domaintype*]
harole -listtypes
harole -listoperations [-type *roletype*]
harole -encodepriv *operation(s)*
harole -decodepriv *permission* -type *roletype*
harole -version
harole [-help]

AVAILABILITY

VRTSvcsonec

DESCRIPTION

Use the `harole` command to display the attributes of roles, and add, define, and delete roles within the VCS One cluster.

A role is a set of privileges. A role with valid privileges can be associated to a user for an object or a set of objects specified by *ouvaluepath*. For example, ServerFarmObjectOperator is an object type predefined role in VCS One. This role can be granted to a user on a cluster object or on an *ouvaluepath*.

A privilege is an ability to perform an operation on an object. The privileges that constitute a role usually apply to the object associated with the role. An important extension of this idea is that roles of type `Object` may contain privileges for all object types contained in a cluster, including groups, resources, systems, and users. Similarly, a role of type `Group` may also contain privileges for the resources contained in the group.

Use the `harole` command to add and delete privileges associated with roles. You can also use it to display role types and their privileges, and the roles currently defined in the VCS One cluster.

In VCS One, roles fall into a combination of categories, depending on whether and how users may display or modify them. VCS One role categories include:

System: roles that are predefined in VCS One.

Hidden: roles that are used internally by VCS One and never listed or displayed.

Removable: roles that may be deleted.

Modifiable: roles that users may modify by adding or deleting privileges.

As examples, the VCSOneClientFarm role is in the System and Modifiable categories, whereas all roles created by VCS One users are in the Removable and Modifiable categories. Users may not create System or Hidden roles or change the category of a role.

The following VCS One predefined roles, which are in the System and Removable categories, cannot be modified with the `harole` command:

FrameAdministrator

FrameManager

FrameOperator

GroupAdministrator

GroupOperator

ResourceAdministrator

ResourceOperator

ServerFarmAdministrator

ServerFarmObjectOperator

SystemAdministrator

SystemOperator

UserAdministrator

UserOperator

The following roles are predefined in VCS One. These roles are in the System and Modifiable categories and therefore cannot be removed with the `harole` command:

ContainerUserFarm

ContainerUserGroup

ServerFarmObjectAdministrator

ServerFarmObjectGuest

VCSOneClientFarm

VCSOneClientFrame

VCSOneClientGroup

VCSOneClientSystem

ZoneUserFarm

ZoneUserGroup

A non-root user who has not run the `halogin` command can execute the `harole` command using the `-user` *`user@domain`* option to execute the command with the privileges of the specified user. When issuing the command, the user must enter the fully qualified domain user name and supply a password when prompted. If necessary, the `-domaintype` option can specify the type of domain against which the user is to be authenticated. Supported domain types include:

- "unixpwd"
- "nt"
- "nis"
- "nisplus"
- "ldap"
- "pam"
- "vx" (Symantec Private Domain)

The domain type, by default, is "vx". The domain type is case sensitive.

See NOTES for how to specify "-" and "%" characters in the command line.

OPTIONS

`-add` *rolename* {`-type` *roletype* | `-inherit` *rolename*} [`-desc` *description*]

Add a role, specifying either a role type, using the `-type` option, or an existing role in the VCS One cluster, using the `-inherit` option.

The arguments for the `-type` option must be a valid role type (*roletype*). Use the `-listtypes` option to see a list of valid role types. Valid role types include:

Object

System

Frame

Group

Resource

User

OT

Notifier

Farm

VObject

Automation

CSG

PFrame

VFrame

The VObject, PFrame, and VFrame role types are for internal use only.

The `-inherit` option specifies that the new role have the same role type and privileges as the role from which it inherits. The role may be inherited from a VCS One predefined role or a user-defined role.

The `-desc` option permits a text description, enclosed in quotation marks, for the added role.

`-delete` *rolename*

Remove a role. Only Removable category roles can be deleted.

`-addpriv` *rolename operation(s)*

Add privileges to an existing role. A role is defined as a set of privileges, each of which provides permission to perform an operation.

Unless the role is in the Modifiable category, you may not add privileges to predefined roles in VCS One.

Only privileges valid for a role may be added to it. Use the command `harole -listoperations -type` *`roletype`* to verify valid privileges. The *operation(s)* argument must include the prefix `O_`, `S_`, `G_`, `R_`, `U_`, `T_`, `N_`, `F_`, `V_`, `A_`, `C_`, and `P_`. For example, to indicate the privilege for the operation to freeze a system, the argument would be S_FreezeSystem. You may specify multiple operations, delimiting them by spaces.

`-delpriv` *`rolename operation(s)`*

Delete privileges associated with a role. Privileges may have been added (using `-addpriv`) or inherited when the role was added.

Unless the role is in the Modifiable category, you may not delete privileges from predefined roles in VCS One.

`-rollback` *`rolename`*

Roll back user-modifiable predefined roles in VCS One. Some roles, such as those predefined roles used by VCS One client daemons (VCSOneClientFarm, for example), may be modified, by deleting or adding privileges. The `-rollback` option returns the value of the set of privileges to the default values.

`-display [-all |` *`role(s)`* `] [-attribute` *`attribute(s)`*`]`

Display the information about one or more roles defined by users and the privileges associated with them. If no roles are specified, all user roles are displayed. The `-all` option displays all user roles and roles in the System category. The `-attribute` option displays information about the specified attribute(s).

`-list [-all] [`*`conditional(s)`*`]`

Displays a list of roles whose values match given conditional statement(s). Conditional statements can take three forms: Attribute=Value, Attribute!=Value, Attribute=~Value. Multiple conditional statements imply AND logic. If no conditional statement is specified, all roles in the VCS One cluster are listed.

Using the `-all` option lists all user roles and roles in the System category.

`-value` *`rolename attribute`*

Display the value of a specified attribute of a specified role.

`-modify` *`rolename attribute value`*

Modify the value of a role's attribute.

`-listtypes`

(Offline) List the current role types. This command option does not require connection with the Policy Master.

`-listoperations [-type` *`roletype`*`]`

(Offline) List operations (privileges) associated with role types. This command option does not require connection with the Policy Master.

`-encodepriv` *`operation(s)`*

(Offline) Encode a list of user-readable operations to a binary representation of the permissions associated for the operations. This command option does not require connection with the Policy Master.

`-decodepriv` *`permission`* `-type` *`roletype`*

(Offline) Decode the integer representing the permissions associated with a set of operation privileges to a user-readable list. This command option does not require connection with the Policy Master.

`-version`

Display the current version of the `harole` command.

`[-help]`

Display the usage for the `harole` command. When you enter the command and an option without arguments, syntax for the specific option displays.

EXAMPLES

Enter the command and an option without arguments to find the usage.

```
harole -add
```

Add a role name and specify its type.

```
harole -add DatabaseAdmin -type System
```

Add a role name, specify its type, and provide a description.

```
harole -add OracleOperator -type System -desc "This is an oracle
operator role"
```

Add a role name, inheriting the role type and privileges from an existing role.

```
harole -add MyServerFarmAdministrator -inherit ServerFarmAdministrator
```

Add a role name, specify the role from which it inherits the type and privileges, and provide a description.

```
harole -add MyUserRole -inherit UserAdministrator -desc "This role
is inherited from the default UserAdministrator role"
```

Add privileges to an existing role.

```
harole -addpriv MyUserRole O_AddUser O_DeleteUser
```

Add privileges to an existing role.

```
harole -addpriv MyServerFarmAdministrator O_AddSystem S_FreezeSystem
G_AddResource R_OfflineResource U_EnableUser
```

Delete privileges from an existing role.

```
harole -delpriv MyUserRole O_DeleteUser
```

Delete privileges from an existing role.

```
harole -delpriv MyServerFarmAdministrator G_AddResource U_EnableUser
```

Display the attributes and values for a role.

```
harole -display SystemAdministrator
```

Display the value of a specific attribute for a role.

```
harole -value SystemAdministrator SystemPrivileges
```

Modify the value of a specific attribute for a role.

```
harole -modify CoGroupAdmin SourceFile /foo
```

Encode a list of user operations to integer(s) representing the permissions.

```
harole -encodepriv U_ModifyUser U_AddPrivilege
```

Automation	:	0
Object	:	0
System	:	0
Frame	:	0
Group	:	0
Resource	:	0
User	:	3
OT	:	0
Notifier	:	0
Farm	:	0
CSG	:	0

Decode the permissions associated with a role type to user-readable list. See the previous example.

```
harole -decodepriv 3 -type user

U_ModifyUser

U_AddPrivilege
```

NOTES

When using the command to specify or modify an attribute's value that begins with a dash ("-"), precede the value with a percent sign ("%"). For example, specify `-y` as `%-y`. Likewise, precede a value that starts with a percent sign with another percent sign.

SEE ALSO

`hauser`(1M), `halogin`(1M)

harule

`harule` – add, delete, modify, enable, disable, or display a rule

SYNOPSIS

```
UNIX: /opt/VRTSvcsone/bin/harule

Windows: %VCSONE_HOME%\bin\harule

harule -add  rule_name object_type ouPath [-user username@domain
-domaintype domaintype]
harule -delete rule_name [-user username@domain -domaintype
domaintype]
harule -modify rule_name attribute_name attribute_value [-user
username@domain -domaintype domaintype]
harule -enable rule_name [-user username@domain -domaintype
domaintype]
harule -disable rule_name [-user username@domain -domaintype
domaintype]
harule -display rule_name [-user username@domain -domaintype
domaintype]
harule -list [-user username@domain -domaintype domaintype]
harule -listevents [-type object type] [-user username@domain
-domaintype domaintype]
harule -value rule_name attribute_name [-user username@domain
-domaintype domaintype]
harule [-help]
harule -version
```

AVAILABILITY

VRTSvcsonew

DESCRIPTION

The `harule` command allows you to add, delete, modify, enable, disable, and list rules. Rules are triggered by a Policy Master event. You can use the `harule` command to display rules and their attributes.

A non-root user who has not run the `halogin` command can execute the `harule` command using the `-user` *user@domain* option to execute the command with

the privileges of the specified user. When issuing the command, the user must enter the fully qualified domain user name and supply a password when prompted. If necessary, the `-domaintype` option can specify the type of domain against which the user is to be authenticated. Supported domain types include:

- "unixpwd"
- "nt"
- "nis"
- "nisplus"
- "ldap"
- "pam"
- "vx" (Symantec Private Domain)

The domain type, by default, is "vx". The domain type is case sensitive.

See NOTES for how to specify "-" and "%" characters in the command line.

OPTIONS

`-add` *`rule_name object_type ouPath`* `[-user` *`username@domain`* `-domaintype` *`domaintype`*`]`

Adds all notification rules. *object_type* Indicates that the rule applies for events for a specific object type (for example, Group, System, Resource, Composite Service Group, PFrame, VFrame, or User). *ouPath* Associates the rule to a particular OU node. The rule cannot access objects outside the scope of the OUPath where it is defined.

`-delete` *`rule_name`* `[-user` *`username@domain`* `-domaintype` *`domaintype`*`]`

Deletes the specified rule.

`-modify` *`rule_name attribute_name attribute_value`* `[-user` *`username@domain`* `-domaintype` *`domaintype`*`]`

Modifies all notification rules. The attributes that you can modify in a rule are description, quiettime, objectselectionvalue, eventselectionvalue, emailrecipients, snmphosts, and sysloghosts.

`-enable` *`rule_name`* `[-user` *`username@domain`* `-domaintype` *`domaintype`*`]`

Enables a single, specified rule.

`-disable` *`rule_name`* `[-user` *`username@domain`* `-domaintype` *`domaintype`*`]`

Disables a single, specified rule. Events cannot trigger a disabled rule.

`-display` *`rule_name`* `[-user` *`username@domain`* `-domaintype` *`domaintype`*`]`

Displays all policy rules and all the attributes for each rule or a single, specific rule. Use the *rule_name* option to display a specific rule.

`-list [-user` *`username@domain`* `-domaintype` *`domaintype`*`]`

Lists notification rules. The command lists the name of the rule, the object type for the rule, and the name of the owner.

`-listevents [-type` *`object type`*`] [-user` *`username@domain`* `-domaintype` *`domaintype`*`]`

Lists the events that the Policy Master can trigger for the specified object type. If you do not specify a -type, then the command lists all events.

`-value` *`rule_name attribute_name`* `[-user` *`username@domain`* `-domaintype` *`domaintype`*`]`

Returns the value of the attribute for a given rule.

`[-help]`

Displays the command usage for `harule`.

`-version`

Displays the version of `harule`.

EXAMPLES

To modify the properties of a rule, use the `-modify` option. For example:

```
# harule -modify EmailRule EmailRecipients "email1@domain.com
email2@domain.com"
VCS One INFO V-97-102-1217 Attribute EmailRecipients
on rule EmailRecipients updated to value email1@domain.com
email2@domain.com
```

To enable a rule, use the `-enable` option. For example:

```
# harule -enable DependencyViolation -user vcsone_admin@sys1
-domaintype unixpw

Password:

VCS One INFO V-97-102-1210 Rule DependencyViolation
successfully enabled.
```

To disable a rule, use the `-disable` option. For example:

```
# harule -disable DependencyViolation -user vcsone_admin@sys1
-domaintype unix
```

```
Password:

VCS One INFO V-97-102-1216 Rule DependencyViolation
successfully disabled.
```

To display the rules and their attributes that apply for a specified user, use the `-display` option. For example:

```
# harule -display -user vcsone_admin@sys1 -domaintype

 unixpwd

Password:
```

#Name	Attribute	Value
ConcurrencyViolation	Creator	simuser@domain
ConcurrencyViolation	Description	Notify in the case of concur
ConcurrencyViolation	EmailRecipients	
ConcurrencyViolation	Enabled	Enabled
ConcurrencyViolation	EventSelectionCriteria	LIST

To display the attributes and attribute values for a specified rule, use the `-display` option. For example:

```
# harule -display DependencyViolation -user vcsone_admin@sys1

-domaintype unix

Password:
```

#Name	Attribute	Value
DependencyViolation	Creator	simuser@domain
DependencyViolation	Description	Notify in the case of depend
DependencyViolation	EmailRecipients	
DependencyViolation	Enabled	Enabled
DependencyViolation	EventSelectionCriteria	LIST

To list the rules that apply for a specified user, use the `-list` option. For example:

```
# harule -list -user vcsone_admin@sys1 -domaintype unixpwd

Password:
```

#Rule	ObjectType	Owner
ConcurrencyViolation	GROUP	simuser@domain
DependencyViolation	GROUP	simuser@domain

To list the events that the Policy Master can trigger for the specified object type, enter the following:

```
# harule -listevents -type GROUP
```

#Event	Severity	Description
GRP-ONLINE	INFORMATION	Service Group Online
GRP_INIT_ONLINE	INFORMATION	Initiated Service Group Online
GRP_OFFLINE	INFORMATION	Service Group Offline
GRP_INIT_OFFLINE	INFORMATION	Initiated Service Group Offline
GRP_FAULT	ERROR	Service Group Fault
GRP_NOFAILOVER	CRITICAL	Service Group Nofailover
GRP_SWITCH	INFORMATION	Service Group Switch
GRP_CONCURRENCY _VIOLATION	CRITICAL	Service Group Concurrency Violation
GRP_KICKOUT	WARNING	Service Group Kicked Out
GRP_SWITCHING_ LOAD_INCREMENTED	ERROR	Service Group switch due to increased load
GRP_ONLINE _CANCELLED_POSSIBLE _CONCURRENCY	CRITICAL	Service Group online cancelled due to possible concurrency
GRP_COMPATIBILITY _VIOLATION	CRITICAL	Service Group Compatibility Violation

GRP_LOAD_VIOLATION	CRITICAL	Service Group Load Violation
GRP_DEPENDENCY _VIOLATION	CRITICAL	Service Group Dependency Violation
GRP_ADD	INFORMATION	Service Group Add
GRP_DELETE	INFORMATION	Service Group Delete
GRP_FREEZE	INFORMATION	Service Group Frozen
GRP_UNFREEZE	INFORMATION	Service Group Unfrozen
GRP_MOVE	INFORMATION	Service Group Moved
GRP_ATTR_CHANGE	INFORMATION	Service Group Changed

To get the value of the attribute for a given rule, enter the following:

```
# harule -value r1 InvalidationReason

[Rule validation failed, Job validation failed, VCS One ERROR
V-97-100-134 SMTP server is not specified.]
```

When using the command to specify or modify an attribute's value that begins with a dash ("-"), precede the value with a percent sign ("%"). For example, specify `-y` as `%-y`. Likewise, precede a value that starts with a percent sign with another percent sign.

SEE ALSO

`hajob`(1M)

haset

`haset` – create and maintain set names

SYNOPSIS

```
UNIX: /opt/VRTSvcsone/bin/haset

Windows: %VCSONE_HOME%\bin\haset

haset -add setname {-ea expression | -ou expression | -ou expression
-ea expression} [-user user@domain -domaintype domaintype]
haset -delete setname [-user user@domain -domaintype domaintype]
haset -display [setname(s)] [-user user@domain -domaintype domaintype]
haset -modify modify_options
haset [-help [-modify]]
haset -version
```

AVAILABILITY

VRTSvcsonec

DESCRIPTION

The `haset` command is used to create and maintain set names. A set name is shorthand for a set, which is a collection of objects specified by an OU expression (*expression*) and/or an EA expression (*expression*). A set name can be used for batch operations on the collection of objects. Use the `haset` command to add and delete set names. You can also use the command to modify a set name and display the associated EA expression (*expression*) and OU expression (*expression*) information for the specified set name.

EA expressions can use the operators `AND` and `OR`. Set expressions are evaluated left to right and there is no operator precedence.

An example of an OU expression is /LOB=Wireline, which is the set of all objects owned by the Wireline LOB.

An example of an EA expression is Architecture=x86 AND OSType=Solaris, which is the set of all Solaris x86 systems.

EA and OU expression strings that contain spaces must be enclosed in quotes.

OPTIONS

`-add` *`setname`* {`-ea` *`expression`* | `-ou` *`expression`* | `-ou` *`expression`* `-ea` *`expression`*} [`-user` *`user@domain`* `-domaintype` *`domaintype`*]

Create a set name with the name specified by *setname*. The set name is defined by the specified *expression*.

An OU expression cannot contain spaces.

An EA expression must be enclosed in double quotes if it contains spaces.

An extended attribute value cannot contain a comma.

In addition, an extended attribute value or validation set cannot contain a single quote (') character. The single quote character serves as a delimiter for the value in an EA expression. However, single quotes can be used to enclose a multiword extended attribute value in an EA expression. For example:

```
hagrp -display -ea "ea1= 'new value' and ea2= 'new value2'"
```

`-delete` *`setname`* [`-user` *`user@domain`* `-domaintype` *`domaintype`*]

Delete a set with the name specified by *setname*.

`-display` [*`setname`*] [`-user` *`user@domain`* `-domaintype` *`domaintype`*]

Display the associated *expression* information for the specified set name *setname*. If no *setname* is specified, then the `-display` option will show all the sets in the user's privilege set.

`-modify` *`modify_options`*

The `-modify` option lets you modify a setname's attributes.

You may modify a scalar attribute's existing value.

You may not use `-modify` to change values already defined for a vector, a keylist, or an association attribute. For vector, keylist, and association attributes, the *modify_options*, which include `-add`, `-delete`, `-update`, or `-delete -keys`, may be used.

Refer to the following list of `-modify` commands. You may display the commands using `haset -help -modify`.

`SCALAR`

```
haset -modify setname attribute value
```

`VECTOR`

Use the following command only when the attribute has no value:

```
haset -modify setname attribute value...
```

For vector attributes that have values defined, only the following operations are allowed:

```
haset -modify setname attribute -add value...
```

```
haset -modify setname attribute -delete -keys
```

Note: You cannot delete an individual element of a VECTOR.

KEYLIST

Use the following command only when the attribute has no value:

```
haset -modify setname attribute {key value}...
```

For keylist attributes that have values defined, only the following operations are allowed.

```
haset -modify setname attribute -add {key value}...
```

```
haset -modify setname attribute -update {key value}...
```

```
haset -modify setname attribute -delete key...
```

```
haset -modify setname attribute -delete -keys
```

ASSOCIATION

Use the following command only when the attribute has no value:

```
haset -modify setname attribute {key value}...
```

For association attributes that have values defined, only the following operations are allowed.

```
haset -modify setname attribute -add {key value}...
```

```
haset -modify setname attribute -update {key value}...
```

```
haset -modify setname attribute -delete key...
```

```
haset -modify setname attribute -delete -keys
```

`-help [-modify]`

Display usage for the `haset` command. When you enter the command and an option without arguments, the usage for the specific option is displayed.

The `-modify` option displays the usage for the `-modify` option. See below for a complete list of the `-modify` options.

`-version`

Display command version information.

EXAMPLES

Add the set name MySolSystems defined by OU and EA expressions.

```
# haset -add MySolSystems -ou /ob=wireline
-ea "Architecture=x86 AND OSType=Solaris"
```

Display the EA expression and OU expression information for a set.

```
# haset -display
```

Modify the set MySolSystems EA expression.

```
# haset -modify MySolSystems EAExpression "Architecture=sparc"
```

Delete the set name MySolSystems.

```
# haset -delete MySolSystems
```

NOTES

When using the command to specify or modify an attribute's value that begins with a dash ("-"), precede the value with a percent sign ("%"). For example, specify `-y` as `%-y`. Likewise, precede a value that starts with a percent sign with another percent sign.

SEE ALSO

`haou`(1M), `haea`(1M)

hasim

`hasim` – start and stop the VCS One Simulator, and simulate faults of systems, pframes, vframes, resources, service groups and clusters from the command line

SYNOPSIS

```
hasim -start [-pm] [-vcsonesim] [-extended [-no_operation]] [-fore]
hasim -stop [-pm] [-vcsonesim]
hasim -faultsys system(s)
hasim -startsys system(s)
hasim -faultpframe pframes(s)
hasim -startpframe pframes(s)
hasim -faultcluster remote_cluster
hasim -killclient {system(s) | pframe(s)}
hasim -faultres resource [-sys system] [-grp group]
hasim -faultres resource [-pframe pframe] [-vframe vframe]
hasim -clearresfault resource {-sys system [-grp group]} | {-pframe
pframe [-vframe vframe]}
hasim -faultgrp group [-sys system]
hasim -faultvframe vframe [-pframe pframe]
hasim -migrate vframe -to pframe
hasim -faultrlink remote_cluster [rlink]
hasim -clearrlinkfault remote_cluster [rlink]
hasim -enablelink system [-hb]
hasim -disablelink system [-hb]
hasim -help
hasim -version
```

AVAILABILITY

vcsonesim

DESCRIPTION

The VCS One Simulator is available for Windows. You can install the VCS One Simulator on one or more Windows systems.

The `hasim` command enables you to simulate faulting systems, pframes, service groups, vframes, and resources to verify and modify configurations in the VCS One cluster in a simulated mode.

When you start the Simulator, you can configure messages from the Simulator to go to stdout instead of the engine log (the default).

Use the `hasim -start` option to start the Simulator, which starts the Policy Master and proxysim (vcsonesim) processes.

To simulate the loss of the Policy Master, you may kill the vcsoned process on the system running the Simulator. After killing the Policy Master daemon process, do not clean the database or load a different configuration. When you restart the Policy Master process, use the `hasim -start` command and use the existing configuration.

The `-disablelink` and `-enablelink` options let you simulate the loss of communications due to hardware failures.

The `-faultcluster`, `-faultrlink`, and `-clearrlinkfault` options let you simulate the fault of a remote cluster or the communication link with a remote cluster.

OPTIONS

`-start [-pm] [-vcsonesim] [-extended [-no_operation]] [-fore]`

Starts the Simulator. The `-pm` option starts only the Policy Master. The `-vcsonesim` option starts the proxysim (vcsonesim) process. The proxysim process can start only if thePolicy Master process is running.

If you specify the `-extended` option, the systems, pframes, vframes, resources, and group's states/istates are retrieved from the database instead of being rediscovered. Use this option if you want to start the Simulator with the same state information that is present in the database. When the Simulator is started with the `-extended` option specified, the systems, resources, and group's states/istates are preserved.

If you specify the `-no_operation` option with the `-extended` option, you will be in read-only mode. You cannot perform write operations. Using the `-no_operation` option is useful for debugging. The systems, pframes, vframes, resources and group's states/istates are preserved and you can see the exact state/istate information for all the objects present in the database.

The `-fore` option specifies messages go to stdout rather than to the VCS One Simulator engine log.

`-stop [-pm] [-vcsonesim]`

Stops the Simulator. The `-pm` option simulates the loss of the Policy Master server. The `-vcsonesim` option simulates a connection fault between all of the VCS One client daemon systems and the Policy Master.

`-faultsys` *`system(s)`*

`-faultsys` Simulates the faulting of a system or systems.

`-startsys` *`system(s)`*

`-startsys` Simulates restoring a faulted system or systems to a RUNNING state.

`-faultpframe` *`pframes(s)`*

Simulates the faulting of a pframe.

`-startpframe` *`pframes(s)`*

Simulates restoring a faulted pframe or pframes to a RUNNING state.

`-faultcluster` *`remote_cluster`*

Use this option from the local cluster to fault the remote cluster. Faulting the cluster means that the Policy Master service group has faulted. Faulting the remote cluster causes the vcsoned process running on it to quit. To use this option, two Simulator instances must be running on the same system. You can invoke the `hasim -faultcluster` command only from the lexically lower cluster.

`-killclient {`*`system(s)`* `|` *`pframe(s)`*`}`

`-killclient` Simulates faulting a VCS One client daemon system or systems.

`-faultres` *`resource`* `[-sys` *`system`*`] [-grp` *`group`*`]`

Use `-faultres` to simulate faulting a resource on a specific system. Use `-grp` to specify a service group. Use `hasim -clearfault` or `hares -clear` to clear the resource fault. If you do not specify a system name, the resource faults on all the systems on which it is online.

`-faultres` *`resource`* `[-pframe` *`pframe`*`] [-vframe` *`vframe`*`]`

Simulates faulting a resource on a specific pframe. use `-vframe` to specify a specific vframe.

`-clearresfault` *`resource`* `{-sys` *`system`* `[-grp` *`group`*`]} | {-pframe` *`pframe`* `[-vframe` *`vframe`*`]}`

Simulate clearing a fault on a specific system or pframe. Use `-grp` to specify a service group. Use `vframe` to specify a vframe. Use `hares -clear` to clear the resource fault.

`-faultgrp` *`group`* `[-sys` *`system`*`]`

Use `-faultgrp` to simulate faulting a service group. You may specify a system. Use the `hagrp -clear` option to clear the service group fault.

`-faultvframe` *`vframe`* `[-pframe` *`pframe`*`]`

Simulates faulting a vframe. You may specify the pframe.

`-migrate` *`vframe`* `-to` *`pframe`*

Simulates the migrate action of a vframe.

`-faultrlink` *`remote_cluster`* `[`*`rlink`*`]`

Use this option from the local cluster to disconnect the link to a remote cluster specified by *rlink*. To use this option, two Simulator instances must be running on the same system. You can invoke the `hasim -faultrlink` command only from the lexically lower cluster.

If you specify a link, the Simulator disconnects that link and changes the link status to DOWN. The link name must be an entry in the NetworkConnections attribute.

If you do not specify a link, the Simulator disconnects the main communication link and changes the link status to DOWN.

When all the links are DOWN, the Simulator changes the state of the remote cluster to FA ULTED.

`-clearlinkfault` *`remote_cluster`* `[`*`rlink`*`]`

Use this option to clear a remote link fault. To use this option, two Simulator instances must be running on the same system. You can invoke the `hasim -clearrlinkfault` only from the lexically lower cluster.

When all the links are DOWN, the state of the remote cluster is FAULTED. When you clear any one remote link fault, the Simulator changes the state of the remote cluster to RUNNING.

If you specify a link, the Simulator connects it and changes the link status to UP. The link name must be an entry in the NetworkConnections attribute.

If you do not specify a link, the Simulator connects the first available link (in the NetworkConnections attribute) that is DOWN and changes the link status to UP.

`-enablelink` *`system`* `[-hb]`

Use this option to enable a disabled link. The command restarts dataflow on the indicated link to simulate an intermittent link. Use the `-hb` option to enable dataflow on the heartbeat link. By default, the Simulator initially creates two links for each simulated system, one for communications and one for heartbeating.

`-disablelink` *`system`* `[-hb]`

Stops dataflow over a link to simulate a hardware failure in the communications path. By default, dataflow is stopped on the communications link. Use the `-hb` option to stop dataflow on the heartbeat link.

By default, the Simulator initially creates two links for each simulated system, one for communications and one for heartbeating.

`-help`

Display usage for the `hasim` command.

`-version`

Display the command version.

EXAMPLES

To simulate the fault of a system, enter:

hasim -faultsys *sys1*

To simulate starting two systems (sys1 and sys2), enter:

```
hasim -startsys sys1 sys2
```

To simulate the "domain down, node active" (DDNA) state of a client (that is, killing the client daemon, leaving the system active), enter:

```
hasim -killclient sys1 sys2
```

To simulate a resource fault on a system (sys1) enter:

```
hasim -faultres res1 -sys sys1
```

To simulate a resource fault on a vframe (vframe1), which is on a pframe, enter:

```
hasim -faultres res1 -vframe vframe1 -pframe pframe1
```

To simulate a resource fault on a group (grp1), enter:

```
hasim -faultres res1 -grp grp1
```

To simulate a resource fault on a group (grp1), which is on a system (sys1) enter:

```
hasim -faultres res1 -sys sys1 -grp grp1
```

To simulate clearing the resource fault on system (sys1), enter:

```
hasim -clearresfault resource -sys sys1
```

To simulate clearing the resource fault on a vframe (vframe1), that is on a pframe (pframe1) enter:

```
hasim -clearresfault resource -vframe vframe1 -pframe pframe1
```

To simulate clearing the resource fault on a group (grp1), enter:

```
hasim -clearresfault resource -sys sys1 -grp grp1
```

To simulate a group fault, enter:

```
hasim -faultgrp grp1
```

To simulate a group fault on a system (sys1), enter:

```
hasim -faultgrp grp1 -sys sys1
```

To simulate a vframe fault on a pframe, enter:

```
hasim -faultvframe vframe1 -pframe pframe1
```

To clear the group fault, enter:

```
hagrp -clear grp1
```

To simulate disabling the heartbeat link (hb) on a system (sys1), enter:

```
hasim -disablelink -hb sys1
```

To simulate disabling both the heartbeat and the data links, enter:

```
hasim -disablelink sys1
```

To simulate enabling only the heartbeat link, enter:

```
hasim -enablelink -hb sys1
```

To simulate enabling both the heartbeat and the data links, enter:

```
hasim -enablelink sys1
```

To simulate a remote cluster fault, enter:

```
hasim -faultcluster remote_cluster
```

To simulate the migration of a vframe (vframe1) to a pframe (pframe2), enter:

```
hasim -migrate vframe1 -to pframe2
```

SEE ALSO

`hagrp`(1M), `hares`(1M), `hasys`(1M)

hastart

`hastart` – start VCS One processes in the VCS One cluster. The VCS One cluster includes the Policy Master daemon, the Policy Master service group (PMSG), and the disaster recovery service group (DRSG), if disaster recovery is configured. The VCS One cluster also includes the VCS process, client daemon (vcsoneclientd) on the Director client daemon systems, configuration database, and web GUI console.

SYNOPSIS

```
UNIX: /opt/VRTSvcsone/bin/hastart

Windows: %VCSONE_HOME%\bin\hastart

hastart -cluster [-cold] [ -manual] [-rthrds Number_of_Threads] [-sys
sys_name]
hastart -db -sys sys_name
hastart -web
hastart -pmm [-onenode]
hastart -pm [-cold ] [-manual] [-rthrds Number_of_Threads]
hastart -client
hastart -version
hastart [-help]
```

AVAILABILITY

VRTSvcsonec

DESCRIPTION

Veritas Cluster Server One provides high availability for the Policy Master Service Group (PMSG) and the disaster recovery service group (DRSG), in the Policy Master cluster. VCS One provides high availability by controlling and monitoring the groups' resources.

Use the `-pmm` option to start VCS from a local system in the Policy Master cluster. The PMSG and the DRSG (if configured) automatically start on one of the Policy Master systems that are included in the PMSG AutoStart list. You may start VCS on one system.

Use the `hastart -cluster` command if VCS is running in the Policy Master cluster, but the PMSG and DRSG (if configured) are not up on any systems. Use the `hastart`

`-cluster` command to start the PMSG and DRSG (if configured). You may specify a system.

With the `-cluster` option, you can start the Policy Master service group or the Policy Master server in the `cold` mode instead of the normal mode. In the normal mode, the Policy Master performs recovery operations. The Policy Master performs those operations based on known state information and on the group transition queue (GTQ) entries. However, if `cold` is specified, the Policy Master performs no recovery operations.

Use the `manual` option to specify that the Policy Master is to wait for user input when reacting to any faults after it has come up. If the `-cold` and `-manual` options are specified, as the Policy Master comes up, it does not perform any recovery operations. When the Policy Master is up, it waits for user input before it reacts to any FAULTS.

The `-pm` and `-db` options provide the means to perform maintenance tasks on the VCS One configuration. You can use the `hastop -pm` command to stop the Policy Master server processes without stopping the other resources in the PMSG. The `hastop -pm` command stops the DRSG before it stops the Policy Master server processes. Or, you can use the `hastop -db` command to stop the database only. You cannot stop the database if the Policy Master is running.

You can restart the database using `hastart -db` and online the Policy Master with `hastart -pm`. You may also start the Policy Master in the normal or the `cold` modes, and specify the `manual` mode in either case.

OPTIONS

`-cluster [-cold ] [-manual] [-rthrds` *`Number_of_Threads`*`] [-sys` *`sys_name`*`]`

Start the Policy Master service group on the local system or a specified system and bring up the VCS One cluster. You may specify a `-cold` startup mode. In a disaster recovery configuration, this option also brings the DRSG online. If you specify the `-manual` startup mode, the Policy Master waits on user input before it reacts to faults. Use the `-rthrds` option to increase the number of threads that service read-only commands in the Policy Master. Doing so can enhance Policy Master performance. By default, the number of threads is 4.

`-db -sys` *`sys_name`*

Start the VCS One database. Specify a system if necessary. The `-db` option is useful when you have stopped the database using the `hastop -db` command for changing of the configuration file or other maintenance action.

After starting the database, start the Policy Master using the `-pm` option.

`-web`

Start the VCS One web GUI console.

In addition, the `hastart -web` command changes the MonitorInterval of the Web server resource to the default, which is 60 seconds.

`-pmm [-onenode]`

Start VCS on each system in the Policy Master cluster. The PMSG starts based on the PMSG AutoStart list.

The `-onenode` option may be used to start VCS on one system for test purposes. LLT and GAB components do not start. Do not use the `-onenode` option in a multinode Policy Master cluster.

`-pm [-cold ] [-manual] [-rthrds` *`Number_of_Threads`*`]`

Start the Policy Master server daemon. In a disaster recovery configuration, this option also brings the DRSG online. If the Policy Master is down, you can start it in the `-cold` mode so that the Policy Master does not attempt to perform recovery. The `-manual` option specifies that, when the Policy Master is up, it waits for user input before it reacts to faults.

Use the `-pm` option if you have stopped the Policy Master using `hastop -pm` to perform a maintenance task.

Use the `-rthrds` option to increase the number of threads that service read-only commands in the Policy Master. Doing so can enhance Policy Master performance. By default, the number of threads is 4.

`-client`

Start the VCS One client daemon (vcsoneclientd) on a local system.

`-version`

Display the version of the `hastart` command.

`[-help]`

Display usage for the `hastart` command.

SEE ALSO

`hastop`(1M), `haadmin`(1M)

hastatus

`hastatus` – display the states of systems, groups, composite service groups, and resources in the VCS One cluster

SYNOPSIS

```
UNIX: /opt/VRTSvcsone/bin/hastatus

Windows: %VCSONE_HOME%\bin\hastatus

hastatus [-sound] [-user user@domain -domaintype domaintype]
hastatus -summary [-sys system] [-user user@domain -domaintype
domaintype]
hastatus [-sound] -grp group(s) [-user user@domain -domaintype
domaintype]
hastatus [-sound] -csg csg(s) [-user user@domain -domaintype
domaintype]
hastatus [-sound] -sys system(s) [-user user@domain -domaintype
domaintype]
hastatus [-sound] -resource resource(s) [-user user@domain -domaintype
domaintype]
hastatus -version
hastatus -help
```

AVAILABILITY

VRTSvcsonec

DESCRIPTION

The `hastatus` command displays group, composite service group, system, and resource status. The command shows either summary information or information for a specific set of objects. The `-sound` option provides an audible alert when faulted objects are displayed.

A non-root user who has not run the `halogin` command can execute the `hastatus` command using the `-user` *`user@domain`* option to execute the command with the privileges of the specified user. When issuing the command, the user must enter the fully qualified domain user name and supply a password when prompted. If necessary, the `-domaintype` option can specify the type of domain against which the user is to be authenticated. Supported domain types include:

- "unixpwd"
- "nt"
- "nis"
- "nisplus"
- "ldap"
- "pam"
- "vx" (Symantec Private Domain)

The domain type, by default, is "vx". The domain type is case sensitive.

OPTIONS

`[-sound] [-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Display the status of all systems, groups, and resources. The `-sound` option specifies that an audible alert, such as a bell sound, occurs when a system, service group, or resource fault appears in displayed output.

`-summary [-sys` *`system`*`] [-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Display a tabular summary of the status of systems (VCS One client systems) service groups, and composite service groups in the VCS One cluster. In a global cluster setup, the `-summary` option also displays the state of the remote clusters.

`[-sound] -grp` *`group(s)`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Report status information for the specified service groups and the resources configured for the service groups. The `-sound` option provides an audible alert for a faulted service group when it appears in the displayed output.

`[-sound] -csg` *`csg(s)`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Report status information for the specified composite service group(s). The `-sound` option provides an audible alert for a composite service group that has the ATTN flag set when it appears in the displayed output.

`[-sound] -sys` *`system(s)`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Report status information for the specified system(s) and for the service groups and resources configured on the system(s). The `-sound` option provides an audible alert for a faulted system when it appears in the displayed output.

`[-sound] -resource` *`resource(s)`* `[-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Report on the state of the specified resource on each system it is configured. The `-sound` option provides an audible alert for a faulted resource appearing in the displayed output.

`-version`

Display command version information.

`-help`

Display usage for the `hastatus` command.

NOTES

You may use the `hastatus` command (except for the `-summary` option) even while the VCS One Policy Master is not running. It will keep on attempting to connect if the Policy Master is not running. As soon as the Policy Master is running, the `hastatus` command output displays.

SEE ALSO

`hagrp`(1M), `hares`(1M), `hacsg`(1M), `halogin`(1M)

hastop

`hastop` – take the VCS One Policy Master service group offline, or stop the Cluster Server (VCS) in the Policy Master base cluster on one or more systems in the VCS One cluster. If disaster recovery is configured, this command also takes the disaster recovery service group offline. You may also use the command to stop the VCS One client daemon or the VCS One web GUI console.

SYNOPSIS

```
UNIX: /opt/VRTSvcsone/bin/hastop

Windows: %VCSONE_HOME%\bin\hastop

hastop -cluster -pm
hastop -cluster -all [-force] [-user user@domain -domaintype
domaintype]
hastop -db
hastop -web
hastop -pm
hastop -pmm -local [-force | -evacuate | -noautodisable]
hastop -pmm -local [-force | -evacuate -noautodisable]
hastop -pmm -sys system(s) [-force | -evacuate | -noautodisable]
hastop -pmm -sys system(s) [-force | -evacuate -noautodisable]
hastop -pmm -all [-force]
hastop -client -local [-force | [-propagate] -evacuate] [-user
user@domain -domaintype domaintype]
hastop -client -local -propagate [-user user@domain -domaintype
domaintype]
hastop -client -sys system(s) [[-actonnodefault] -force | [-propagate]
-evacuate] [-user user@domain -domaintype domaintype]
hastop -client -sys system(s) -propagate [-user user@domain
-domaintype domaintype]
hastop -client -all [-force] [-user user@domain -domaintype
domaintype]
hastop -client -pframe pframe(s) [[-actonnodefault] -force |
[-propagate] -evacuate] [-user user@domain -domaintype domaintype]
hastop -client -pframe pframe(s) -propagate [-user user@domain
-domaintype domaintype]
hastop -version
hastop [-help]
```

AVAILABILITY

VRTSvcsonec

DESCRIPTION

The `hastop` utility with the `-cluster -pm` options stops the Policy Master server daemon by taking the Policy Master service group (PMSG) offline in the Policy Master cluster, which runs Cluster Server to provide high availability for the Policy Master server. Taking the PMSG offline also stops all resources in the Policy Master service group, including the storage and the database. In a disaster recovery configuration, this option also stops the disaster recovery service group (DRSG).

You can use the command to stop the Policy Master server temporarily for maintenance or similar reasons. Stopping the Policy Master server using `hastop` allows the VCS One client daemons and service groups on client systems to continue running. However, while the Policy Master server daemon is not running, there is no high availability in the VCS One cluster. In a disaster recovery configuration, the communication with the remote cluster is terminated.

The `hastop` utility with the `-client` option stops the VCS One client daemon on specified systems or on all systems in the VCS One cluster.

Veritas Cluster Server (VCS) provides high-availability for the PMSG and the DRSG (if configured) by controlling and monitoring their resources. The `-pmm` option enables administrators to stop VCS on a specific system or on all systems in the base cluster.

The `-force` option provides the ability to stop the daemon on a system while keeping the service groups online. The `-evacuate` option provides the ability to migrate the service groups to other systems when stopping the daemon on a specific system. When administrators are sure the service group is not online elsewhere, they may use the `-noautodisable` option to specify that the group may be brought online.

A non-root user who has not run the `halogin` command can execute the `hastop` command using the `-user` *`user@domain`* option to execute the command with the privileges of the specified user. When issuing the command, the user must enter the fully qualified domain user name and supply a password when prompted. If necessary, the `-domaintype` option can specify the type of domain that the user will be authenticated against. Supported domain types include:

- "unixpwd"
- "nt"
- "nis"

- "nisplus"
- "ldap"
- "pam"
- "vx" (Symantec Private Domain)

The domain type, by default, is "vx". The domain type is case sensitive.

OPTIONS

`-cluster -pm`

Take the Policy Master service group (PMSG) offline. All resources in the group are taken offline. In a disaster recovery configuration, this option also takes the DRSG offline.

`-cluster -all [-force]`

Stop all instances of the VCS One client daemon. When all VCS One client daemons are stopped, take the Policy Master service group offline. In a disaster recovery configuration, this option also takes the DRSG offline. Use `-force` to keep applications running.

`-db`

Take the VCS One database resource offline.

`-web`

Stop the VCS One web GUI console.

In a disaster recovery configuration, when you stop the VCS One Web GUI console using the`-web` option, the `hastop` command kills the Tomcat server instead of taking the VCS resource offline. This behavior occurs due to a group dependency between the PMSG and the DRSG. The status of the VCSOneweb resource continues to be ONLINE. To verify that the Web server has been killed, use the following command:

```
ps -ef|grep java
```

In addition, the `hastop -web` command changes the MonitorInterval of the Web server resource to one week.

After the Web GUI console has stopped, re-start it using the `hastart -web` command before switching the PMSG to another system in the Policy Master cluster.

`-pm`

Stops the VCS One Policy Master daemon.

In a disaster recovery configuration, this option also takes the DRSG offline.

`-pmm -local [-force | -evacuate | -noautodisable | -evacuate -noautodisable]`

Use the `-pmm -local` option to stop the Cluster Server on the current system. Use the `-force`, `-evacuate`, `-noautodisable`, or `-evacuate -noautodisable` options as needed.

`-pmm -sys` *`system(s)`* `[-force | -evacuate | -evacuate -noautodisable]`

Use the `-pmm -sys` option to stop the Cluster Server on one or more specified systems. Use the `-force`, `-evacuate`, `-noautodisable`, or `-evacuate -noautodisable` options as needed.

`-force`

Use the `-force` option to specify that service groups running on the system continue to run.

`-evacuate`

Use the `-evacuate` option to specify that the service groups be migrated to other systems.

`-noautodisable`

Use the `-noautodisable` option to specify that the service group be brought online elsewhere in the cluster without probing.

`-pmm -all [-force]`

Stop the Cluster Server in the base cluster on all systems.

`-client -local [-force | [-propagate] -evacuate]`

Stop the VCS One client daemon on the local system. Use the `-force` or `-evacuate` option as needed. When the `-propagate` option is used with `-evacuate`, it brings online the service groups on the system and any global parent service groups on other systems. If `-evacuate` is not used, the command takes offline all the service groups on the system as well as the global parents that are online elsewhere.

`-client -local -propagate`

Stop the VCS One client daemon on the local system. This command option takes offline all the service groups on the local system as well as the global parent service groups that are online on other systems. To run this command, you must have the OFFLINE privilege for the global parent service groups.

`-client -sys` *`system(s)`* `[[-actonnodefault] -force | [-propagate] -evacuate]`

Stop the VCS One client daemon on one or more specified systems. Use the `-actonnodefault`, `-force`, and `-evacuate` options as needed. When the `-propagate` option is used with `-evacuate`, it brings online the service groups

on the system and any global parent service groups on other systems. If `-evacuate` is not used, the command takes offline all the service groups on the system as well as the global parents that are online elsewhere.

Under normal circumstances, if a system faults after the VCS One client is stopped, service groups that are online on the system do not fail over. If failover of these service groups is required, use the `actonnodefault` option. The `actonnodefault` option causes service groups that are online when the VCS One client stops to fail over.

`-client -sys` *`system(s)`* `-propagate`

Stop the VCS One client daemon on the specified system(s). This command option takes offline all the service groups on the specified system(s) as well as the global parent service groups that are online on other systems. To run this command, you must have the OFFLINE privilege for the global parent service groups.

`-client -all [-force]`

Stop the VCS One client daemon on all systems. Use the `-force` option as needed.

`-client -pframe` *`pframe(s)`* `[[-actonnodefault] -force | [-propagate] -evacuate] [-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Stop the VCS One client daemon on one or more specified pframes. Use the`-actonnodefault`, `-force`, and `-evacuate` options as needed. When the`-propagate` option is used with `-evacuate`, it brings online the vframes on the pframe and any global parent vframes on other pframes. If `-evacuate` is not used, the command takes offline all the vframes on the pframe as well as the global parents that are online elsewhere.

Under normal circumstances, if a pframe faults after the VCS One client is stopped, vframes that are online on the pframe do not fail over. If failover of these vframes is required, use the `-actonnodefault` option.
The`-actonnodefault` option causes vframes that are online when the VCS One client stops to fail over.

`hastop -client -pframe` *`pframe(s)`* `-propagate [-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Stop the VCS One client daemon on the specified pframe(s). This command option takes offline all the vframes on the specified pframe(s) as well as the global parent vframes that are online on other pframes. To run this command, you must have the OFFLINE privilege for the global parent vframes.

`-version`

Display the command version.

`[-help]`

Display usage for the `hastop` command.

SEE ALSO

`hastart`(1M), `haadmin`(1M)

hasys

`hasys` – add, modify, or delete a system, and display or list information about systems

SYNOPSIS

UNIX: /opt/VRTSvcsone/bin/hasys

Windows: %VCSONE_HOME%\bin\hasys

hasys -add *system* [-platform *platform*] [ouvaluepath] [-user *username@domain* -domaintype *domaintype*]
hasys -delete *system* [-user *username@domain* -domaintype *domaintype*]
hasys -move [-updateroles] [-refreshvars] *system(s)* -ou *ouvaluepath* [-user *username@domain* -domaintype *domaintype*]
hasys -freeze [-evacuate] {*system(s)* | -ou *ouexpression* [-info] | -ea *eaexpression* [-info] | -ou *ouexpression* -ea *eaexpression* [-info] | -setname *setname* [-info]} [-user *username@domain* -domaintype *domaintype*]
hasys -unfreeze {*system(s)* | -ou *ouexpression* [-info] | -ea *eaexpression* [-info] | -ou *ouexpression* -ea *eaexpression* [-info] | -setname *setname* [-info]} [-user *username@domain* -domaintype *domaintype*] [-info]
hasys -display [*system(s)* | -ou *ouexpression* | -ea *eaexpression* | -ou *ouexpression* -ea *eaexpression* | -setname *setname*] [-attribute *attribute(s)*] [-user *username@domain* -domaintype *domaintype*]
hasys -displayea [*system(s)*] [-attribute *attribute(s)*] [-user *username@domain* -domaintype *domaintype*]
hasys -enablevmha *system* [-user *username@domain* -domaintype *domaintype*]
hasys -disablevmha *system* [-user *username@domain* -domaintype *domaintype*]
hasys -list [*conditional(s)*] [-user *username@domain* -domaintype *domaintype*]
hasys -clientversion [*system(s)*] [-user *username@domain* -domaintype *domaintype*]
hasys -state [*system(s)* | -ou *ouexpression* | -ea *eaexpression* | -ou *ouexpression* -ea *eaexpression* | -setname *setname*] [-user *username@domain* -domaintype *domaintype*]

```
hasys -value system attribute [-user username@domain -domaintype
domaintype]
hasys -infovars system attribute [key] [-user username@domain
-domaintype domaintype]
hasys -nodeid [nodeid] [-user username@domain -domaintype domaintype]
hasys -fault system [-user username@domain -domaintype domaintype]
hasys -wait system [-ea] attribute value [-time seconds] [-user
username@domain -domaintype domaintype]
hasys -readconfig system [-user username@domain -domaintype
domaintype]
hasys -modify modify_options
hasys [-help [-modify | -list]]
hasys -version
```

AVAILABILITY

VRTSvcsonec

DESCRIPTION

The `hasys` command allows administrators to manage information about each system. (A system is a node that runs or will run the vcsoneclient daemon.)

For the `-platform` option, supported values for *platform* are:

- aix
- aix/rs6000 (alias aix)
- esx
- hpux
- linux/x86 (alias linux)
- solaris
- solaris/x86
- solaris/sparc (alias solaris)
- windows
- windows/x86

For VMware ESX Server, use `linux` as the platform. Use the explicit platform name where no alias is defined. When *platform* appears in any displays, the full name and not the alias is shown.

A non-root user who has not run the `halogin` command can execute the `hasys` command using the `-user` *`user@domain`* option to execute the command with the privileges of the specified user. When issuing the command, the user must enter the fully qualified domain user name and supply a password when prompted. If necessary, the `-domaintype` option can specify the type of domain against which the user is to be authenticated. Supported domain types include:

- "unixpwd"
- "nt"
- "nis"
- "nisplus"
- "ldap"
- "pam"
- "vx" (Symantec Private Domain)

The domain type, by default, is "vx". The domain type is case sensitive.

See NOTES for how to specify "-" and "%" characters in the command line.

OPTIONS

`-add` *`system`* [`-platform` *`platform`*] [*`ouvaluepath`*] [`-user` *`username@domain`* `-domaintype` *`domaintype`*]

Add a system to the VCS One configuration.

You may optionally specify the *platform* and *ouvaluepath*. Use the `-platform` *`platform`* option to specify the platform for the system. The accepted values for *platform* are aix, aix/rs6000, linux, linux/x86, solaris, solaris/x86, solaris/sparc, windows, and windows/x86. If a default platform has not been set for the VCS One cluster, then you must specify the platform using `-platform` when creating the group. If the DefaultPlatform attribute has been set for the VCS One cluster, it will be used by default for a new system unless you specify the platform using `-platform`.

If you do not specify an OUValuePath (*ouvaluepath*), the system is added to the root (/) of the Organization Tree.

The physical computer represented by this object does not need to exist or be a part of the cluster when the command is issued. The system specified by *system* does not need to correspond to the host name of the actual system, but it is recommended that you match the system with the hostname. If security is enabled, it is almost essential that *system* matches the fully qualified host name of the system in question.

`-delete` *`system`* `[-user` *`username@domain`* `-domaintype` *`domaintype`*`]`

Delete a system from the configuration. The system must not be running the VCS One client daemon. Use `hastop -sys` to stop the VCS One client daemon on the system.

`-move [-updateroles] [-refreshvars]` *`system(s)`* `-ou` *`ouvaluepath`* `[-user` *`username@domain`* `-domaintype` *`domaintype`*`]`

Move a specified system or systems in the VCS One configuration.

Moving a system can cause the system to move outside of a user's home node. In this situation, use the `-updateroles` option. This option deletes the system from the user's role so that the user no longer has privileges on that system. If you do not specify `-updateroles` in this situation, the system move is not allowed.

If you attempt to move a system and if the current value of any of its extended attributes (which is used as resource variable) changes at the new location, the move is rejected. To override this behavior and move the system, use `-refreshvars`. Doing so will modify the value of the resource attributes that use the variable.

`-freeze [-evacuate] [`*`system(s)`* `| -ou` *`ouexpression`* `| -ea` *`eaexpression`* `| -ou` *`ouexpression`* `-ea` *`eaexpression`* `| -setname` *`setname`*`] [-user` *`username@domain`* `-domaintype` *`domaintype`*`] [-info]`

Freeze a system or multiple systems specified by an OU expression (*ouexpression*) and/or an EA expression (*eaexpression*), or set (*setname*). No group configured on the frozen system can come online, whether manually, by failover, or by switching until the system is thawed with the `-unfreeze` option. Using the `-evacuate` option specifies that all groups are switched before the system is frozen; if no other system is available for a service group, it is taken offline. Groups running on other systems do not fail over to a frozen system.

Use the `-info` option to display the objects that the command will act upon if executed. When `-info` is specified, the command is not executed; only information is displayed.

`-unfreeze [`*`system(s)`* `| -ou` *`ouexpression`* `| -ea` *`eaexpression`* `| -ou` *`ouexpression`* `-ea` *`eaexpression`* `| -setname` *`setname`*`] [-user` *`username@domain`* `-domaintype` *`domaintype`*`] [-info]`

Unfreeze a system or multiple systems specified by an OU expression (*ouexpression*) and/or an EA expression (*eaexpression*), or set (*setname*).

`-display [system(s) | -ou ouexpression | -ea eaexpression | -ou ouexpression -ea eaexpression | -setname setname] [-attribute attribute(s)] [-user username@domain -domaintype domaintype]`

Display the attribute names and their values for a specified system or systems specified by a setname or by an *ouexpression* and/or an *eaexpression*. If no system is specified, the attributes and values for all systems are displayed.

An OU expression cannot contain spaces.

An EA expression must be enclosed in double quotes if it contains spaces.

An extended attribute value cannot contain a comma.

In addition, an extended attribute value or validation set cannot contain a single quote (') character. The single quote character serves as a delimiter for the value in an EA expression. However, single quotes can be used to enclose a multiword extended attribute value in an EA expression. For example:

```
hasys -display -ea "ea1= 'new value' and ea2= 'new value2'"
```

`-displayea [system(s)] [-attribute attribute(s)] [-user username@domain -domaintype domaintype]`

Display the extended attributes and their values for a specified system or systems. If no system is specified, the extended attributes and values for all systems are displayed.

`-enablevmha system [-user username@domain -domaintype domaintype]`

Enables VMHA policy for the system. Enable VMHA policy for a system that is linked to a vframe. When VMHA policy is enabled, the Policy Master:

- Does not allow service groups to span systems. Service groups configured on the system cannot have any other system in their SystemList.
- Automatically onlines a vframe when a user onlines a service group configured on the system.
- Restarts or fails over a vframe in response to a service group fault if`PropagateFaultPolicy` is set to Propagate for a service group.
- Sets the capacity of the system equal to the load of the linked vframe (the sum of the Load of all service groups configured on the system).
- Sets the priority of a linked vframe equal to the highest priority of service groups configured on a system. For example:
  If g1(Pri 1) and g2(Pri 5) are configured on the system, then the priority of the linked vframe = 1.

`-disablevmha` *`system`* `[-user` *`username@domain`* `-domaintype` *`domaintype`*`]`

Disables VMHA policy for the system.

`-list [`*`conditional(s)`*`]`

Displays a list of systems whose values match given conditional statement(s). Conditional statements can take three forms: Attribute=Value, Attribute!=Value, Attribute=~Value. Multiple conditional statements imply AND logic. The command lists all systems in the cluster when no conditional statement is used.

For example, `hasys -list PlatformName=linux` lists all the systems where the PlatformName attribute value contains `linux`.

`-clientversion [system(s)] [-user` *`username@domain`* `-domaintype` *`domaintype`*`]`

Displays the version of the client daemon that is installed on the system.

`-state [`*`system(s)`* `| -ou` *`ouexpression`* `| -ea` *`eaexpression`* `| -ou` *`ouexpression`* `-ea` *`eaexpression`* `| -setname` *`setname`*`] [-user` *`username@domain`* `-domaintype` *`domaintype`*`]`

Display the current state of the specified system(s). An OU expression (*ouexpression*) and/or an EA expression (*eaexpression*), or a set (*setname*) may be used to specify systems. The command displays states of all systems if a system or systems are not specified.

`-value` *`system attribute`*

The `-value` option provides the value of a single system attribute. For example, `hasys -value sysb SysState` displays the value of the `SysState` attribute for system `sysb`. Use the `-value` option to show the value of one specific attribute rather than a table of many attribute values shown with the `-display` option.

`-infovars` *`system attribute`* `[`*`key`*`] [-user` *`user@domain`* `-domaintype` *`domaintype`*`]`

Displays the resource attributes that use the specified attribute as a variable. See EXAMPLES.

`-nodeid [`*`nodeid`*`]`

Return the node name and nodeid values for the specified system. Values for the current system are returned if *nodeid* is not provided.

`-fault` *`system`* `[-user` *`username@domain`* `-domaintype` *`domaintype`*`]`

Can be used to force the client to a FAULTED state if it is in the DDNA state. The `-fault` option cannot be used if the client system is in the RUNNING state.

`-wait` *system* `-ea` *attribute value* [`-time` *seconds*]

The `-wait` option is for use in scripts to direct the `hasys` command to wait until the value of the attribute has changed as specified, or until the duration specified by *seconds* has been reached. *seconds* is an integer specifying seconds. If *seconds* is not specified, `hasys` waits indefinitely.

Use the `-ea` option to direct the `hasys` command to wait until the value of an extended attribute changes to the specified value.

The `-wait` option can be used only with changes to scalar attributes.

See EXAMPLES.

`-readconfig` *system* [`-user` *username@domain* `-domaintype` *domaintype*]

The `-readconfig` option allows you to reset the configuration without restarting the VCS One client. Changing only the SystemIPAddrs attribute value is supported. The `-readconfig` option forces the VCS One client daemon (vcsoneclientd) in the RUNNING state to reload the SystemIPAddrs attribute value from the /etc/VRTSvcsone/vcsone.conf file. For example, if a system gets a new IP address, you can edit the SystemIPAddrs entry in the configuration file and then issue this command.

`-modify` *modify_options*

The `-modify` option lets you modify a system's attributes. Some attributes are internal to VCS One and cannot be modified. You can modify any attribute that can be configured in main.xml.

You may modify a scalar attribute's existing value.

You may not use `-modify` to change values already defined for a vector, a keylist, or an association attribute. For vector, keylist, and association attributes, use the *modify_options*, which include `-add`, `-delete`, `-update`, or `-delete -keys`.

Refer to the following list of permissible `-modify` commands. You may display the commands by using `-hasys -help -modify`.

`SCALAR`

`hasys -modify [-refreshvars]` ***system attribute value*** [`-user` *username@domain* `-domaintype` *domaintype*]

If you attempt to modify an extended attribute value that is a variable, an error message is displayed and the value is not modified. To override this behavior and modify an extended attribute value that is a variable, use the `-refreshvars` option. Doing so will modify the value of the resource attributes that use the variable.

VECTOR

Use the following command only when the attribute has no value:

```
hasys -modify system attribute value ... [-user
username@domain -domaintype domaintype]
```

For vector attributes that have values defined, only the following operations are allowed.

```
hasys -modify system attribute -add value ... [-user
username@domain -domaintype domaintype]
```

```
hasys -modify system attribute -delete -keys [-user
username@domain -domaintype domaintype]
```

Note: You cannot delete an individual element of a VECTOR.

KEYLIST

Use the following command only when the attribute has no value:

```
hasys -modify system attribute key ... [-user username@domain
-domaintype domaintype]
```

For keylist attributes that have values defined, only the following operations are allowed.

```
hasys -modify system attribute -add key ... [-user
username@domain -domaintype domaintype]
```

```
hasys -modify system attribute -delete key ... [-user
username@domain -domaintype domaintype]
```

```
hasys -modify system attribute -delete -keys [-user
username@domain -domaintype domaintype]
```

ASSOCIATION

Use the following command only when the attribute has no value:

```
hasys -modify system attribute {key value} ... [-user
username@domain -domaintype domaintype]
```

For association attributes that have values defined, only the following operations are allowed.

```
hasys -modify system attribute -add {key value} ... [-user
username@domain -domaintype domaintype]
```

```
hasys -modify system attribute -update {key value} ... [-user
username@domain -domaintype domaintype]
```

```
hasys -modify system attribute -delete key ... [-user
username@domain -domaintype domaintype]

hasys -modify system attribute -delete -keys [-user
username@domain -domaintype domaintype]
```

`[-help [-modify|-list]]`

The `-help` option displays the command usage for `hasys`. The `-modify` option displays the usage for the `-modify` option. The `-list` option displays the usage for the `-list` option. When you enter the command and an option without arguments, syntax for the specific option displays.

`-version`

Display the version of `hasys`.

EXAMPLES

Example 1. To display the usage syntax for a specific command option, enter the command and an option without arguments. For example, enter:

```
# hasys -value
```

Example 2. From a script, to use the `-wait` option to direct the `hasys` command to block until system `S1` goes into the `RUNNING` state, enter:

```
# hasys -wait S1 SysState RUNNING
```

Example 3. To display all the resource attributes for system `S1` that use *SysInfo:OsVersion* as a variable, enter:

```
# hasys -infovars S1 SysInfo OsVersion
```

If a system name is not specified, information regarding all systems is displayed.

If an attribute name is not specified, information regarding all system attributes is displayed.

When using the command to specify or modify an attribute's value that begins with a dash ("-"), precede the value with a percent sign ("%"). For example, specify `-y` as `%-y`. Likewise, precede a value that starts with a percent sign with another percent sign.

SEE ALSO

`halogin`(1M), `haconf`(1M), `haclus`(1M)

hatype

`hatype` – add, modify, delete, display, or list information about a resource type

SYNOPSIS

```
UNIX: /opt/VRTSvcsone/bin/hatype

Windows: %VCSONE_HOME%\bin\hatype

hatype -add type [-platform platform] [-user user@domain -domaintype
domaintype]
hatype -delete type [-platform platform] [-user user@domain
-domaintype domaintype]
hatype -display [type(s)] [-platform {platform | all}] [-attribute
attribute(s)] [-user user@domain -domaintype domaintype]
hatype -list [conditional(s)] [-platform platform] [-user user@domain
-domaintype domaintype]
hatype -value type attribute [-platform platform] [-user user@domain
-domaintype domaintype]
hatype -resources type [-platform {platform|all}] [-user user@domain
-domaintype domaintype]
hatype -modify modify_options
hatype [-help [-modify | -list]]
hatype -version
```

AVAILABILITY

VRTSvcsonec

DESCRIPTION

The `hatype` command manages information about the various types. For example, it enables you to display and modify static attributes. Each resource that makes up a service is of a specific type, such as a volume or an IP address. Types give VCS One a way to understand how to manage the individual resources. Their management depends entirely on the characteristics of the type.

For the `-platform` option, supported values for *platform* are:

- aix
- aix/rs6000 (alias aix)

- esx
- hpux
- linux
- linux/x86 (alias linux)
- solaris
- solaris/x86
- solaris/sparc (alias solaris)
- windows
- windows/x86

Use the explicit platform name where no alias is defined. When *platform* appears in any displays, the full name and not the alias is shown.

A non-root user who has not run the `halogin` command can execute the `hatype` command using the `-user` *`user@domain`* option to execute the command with the privileges of the specified user. When issuing the command, the user must enter the fully qualified domain user name and supply a password when prompted. If necessary, the `-domaintype` option can specify the type of domain against which the user is to be authenticated. Supported domain types include:

- "unixpwd"
- "nt"
- "nis"
- "nisplus"
- "ldap"
- "pam"
- "vx" (Symantec Private Domain)

The domain type, by default, is "vx". The domain type is case sensitive.

See NOTES for how to specify "-" and "%" characters in the command line.

OPTIONS

`-add` *`type`* [`-platform` *`platform`*]

Add a resource type to the VCS One configuration.

`-delete` *`type`* [`-platform` *`platform`*]

Delete a resource type from the VCS One configuration.

`-display [type(s)] [-attribute attribute(s)] [-platform {platform | all}]`

Display a resource type or all types if none is specified. To display specific attributes, specify them using the `-attribute` option. You may specify a particular platform using the `-platform` option. To get information about the resource type on all platforms, use `-platform all`. If the DefaultPlatform cluster-level attribute is set, you do not need to specify the `-platform` option if the type information is the same as that specified in DefaultPlatform.

`-list [conditional(s)] [-platform platform]`

Displays a list of types whose values match given conditional statement(s). Conditional statements can take three forms: Attribute=Value, Attribute!=Value, Attribute=~Value. Multiple conditional statements imply AND logic. If no conditional statement is specified, all types in the cluster are listed.

`-value type attribute [-platform platform]`

The `-value` option displays the value of a single type attribute. For example, `hatype -value Mount NameRule` displays the value of the NameRule attribute for the Mount type. The `-value` option is used instead of the `-display` option when one specific attribute value is needed rather than a table of many attribute values.

`-resources type [-platform {platform|all}]`

Display a list of resources of the specified resource type. You may specify a particular platform using the `-platform` option. To get information about the resources on all platforms, use `-platform all`. If the DefaultPlatform cluster-level attribute is set, you do not need to specify the`-platform` option if the type information is the same as that specified in DefaultPlatform.

`-modify modify_options`

The `-modify` option lets you modify a type's attributes. Some attributes are internal to VCS One and cannot be modified. You can modify any attribute that can be configured in main.xml.

You may modify a scalar attribute's existing value.

You may not use `-modify` to change values already defined for a vector, a keylist, or an association attribute. For vector, keylist, and association attributes, use the *modify_options*, which include `-add`, `-delete`, `-update`, or `-delete -keys`.

Refer to the following list of permissible `-modify` commands. You may display the commands by using `hatype -help -modify`.

SCALAR

```
hatype -modify type [-platform platform] attr value [-user
user@domain -domaintype domaintype]
```

VECTOR

Use the following command only when the attribute has no value:

```
hatype -modify type attr value ... [-platform platform] [-user
user@domain -domaintype domaintype]
```

For vector attributes that have values defined, only the following operations are allowed.

```
hatype -modify type attr -add value ...[-platform
platform][-user user@domain -domaintype domaintype]
```

```
hatype -modify type attr -delete -keys [-platform platform]
[-user user@domain -domaintype domaintype]
```

Note: You cannot delete an individual element of a VECTOR.

KEYLIST

Use the following command only when the attribute has no value:

```
hatype -modify type attr key ... [-platform platform] [-user
user@domain -domaintype domaintype]
```

For keylist attributes that have values defined, only the following operations are allowed.

```
hatype -modify type attr -add key ... [-platform platform]
[-user user@domain -domaintype domaintype]
```

```
hatype -modify type attr -delete key ... [-platform platform]
[-user user@domain -domaintype domaintype]
```

```
hatype -modify type attr -delete -keys [-platform platform]
[-user user@domain -domaintype domaintype]
```

ASSOCIATION

Use the following command only when the attribute has no value:

```
hatype -modify type attr {key value} ... [-platform platform]
[-user user@domain -domaintype domaintype]
```

For association attributes that have values defined, only the following operations are allowed.

```
hatype -modify type attr -add {key value} ... [-platform
platform] -user user@domain -domaintype domaintype]

hatype -modify type attr -update {key value} ... [-platform
platform] [-user user@domain -domaintype domaintype]

hatype -modify type attr -delete key ... [-platform platform]
[-user user@domain -domaintype domaintype]

hatype -modify type attr -delete -keys [-platform platform]
[-user user@domain -domaintype domaintype]
```

`-help [-modify | -list]`

Display information about using `hatype`. When you enter the command and an option without arguments, syntax for the specific option displays.

The `-modify` option provides `modify`-specific help and the `-list` option provides `-list`-specific help.

`-version`

Display the command version.

EXAMPLES

To display the usage syntax for a specific command option, enter the command and an option without arguments. For example, enter:

```
# hatype -value
```

NOTES

When using the command to specify or modify an attribute's value that begins with a dash ("-"), precede the value with a percent sign ("%"). For example, specify `-y` as `%-y`. Likewise, precede a value that starts with a percent sign with another percent sign.

SEE ALSO

`haattr`(1M), `hares`(1M), `harole`(1M)

hauser

`hauser` – add and remove VCS One users and manage their privileges by assigning them roles

SYNOPSIS

```
UNIX: /opt/VRTSvcsone/bin/hauser

Windows: %VCSONE_HOME%\bin\hauser

hauser -add [-usergroup] user@domain [ouvaluepath] [-user user@domain
-domaintype domaintype]
hauser -delete {-prefs [-all] | [-usergroup user@domain]} [-user
user@domain -domaintype domaintype]
hauser -move [-updateroles] [-usergroup] user@domain(s) -ou
ouvaluepath [-user user@domain -domaintype domaintype]
hauser -enable [-usergroup] user@domain [-user user@domain -domaintype
domaintype]
hauser -disable [-usergroup] user@domain [-user user@domain
-domaintype domaintype]
hauser -addrole [-usergroup] user@domain role_name [-usergroup]
{object(s) | -ou ouvaluepath} [-user user@domain -domaintype
domaintype]
hauser -deleterole [-usergroup] user@domain role_name [-usergroup]
{object(s) | -ou ouvaluepath} [-user user@domain -domaintype
domaintype]
hauser -display [-sys | -usergroup] [user@domain(s) | -ou ouvaluepath]
[-attribute attribute(s)] [-user user@domain -domaintype domaintype]
hauser -display -prefs [user@domain(s) | -all | -ou ouvaluepath]
[-attribute attribute(s)] [-user user@domain -domaintype domaintype]
hauser -value [-sys | -usergroup | -prefs] user@domain attribute
[-user user@domain -domaintype domaintype]
hauser -list [-sys | -usergroup] [conditional(s)] [-user user@domain
-domaintype domaintype]
hauser -list -prefs [-all] [conditional(s)] [-user user@domain
-domaintype domaintype]
hauser -modify modify_options
hauser [-help [-modify | -list ]]
hauser -version
```

AVAILABILITY

VRTSvcsonec

DESCRIPTION

Administrators can use the `hauser` command to add (`-add`) a new user and delete (`-delete`) an existing user in a VCS One cluster. The command can also be used to add and delete usergroups.

Administrators can assign a role (*role_name*) to a user with the `-addrole` option and specify which objects or OUValuePath (*ouvaluepath*) the role applies to. Roles can be created using the `harole` command. Roles are collections of privileges to view, perform operations on, or configure VCS One objects. A user may have multiple roles, the union of which constitutes the user's effective privileges.

An administrator can delete a role previously assigned to a user.

The `enable` and `disable` options allow administrators to change the privilege status of users.

The `display` and `list` commands allow administrators to list users and display information about them.

A non-root user who has not run the `halogin` command can execute the `hauser` command using the `-user` *`user@domain`* option to execute the command with the privileges of the specified user. When issuing the command, the user must enter the fully qualified domain user name and supply a password when prompted. If necessary, the `-domaintype` option can specify the type of domain against which the user is to be authenticated. Supported domain types include:

- "unixpwd"
- "nt"
- "nis"
- "nisplus"
- "ldap"
- "pam"
- "vx" (Symantec Private Domain)

The domain type, by default, is "vx". The domain type is case sensitive.

See NOTES for how to specify "-" and "%" characters in the command line.

OPTIONS

`-add [-usergroup] user@domain [ouvaluepath] [-user user@domain -domaintype domaintype]`

Add a VCS One user by specifying the user's name with *user@domain*. You may also add a usergroup, using the `-usergroup` option. (Use *user@domain* to specify the usergroup name when adding a usergroup.)

You may specify an additional OUValuepath (*ouvaluepath*) to add the user or usergroup to the Organization Tree.

`-delete {-prefs [-all] | [-usergroup user@domain} [-user user@domain -domaintype domaintype]`

Delete a VCS One user by specifying the user's name. You may also delete a usergroup, using the `-usergroup` option. (Use *user@domain* to specify the usergroup name when deleting a usergroup.) You may also delete preferences. Use the `-prefs` option to delete all preferences for the user issuing the command. Use `-prefs -all` to delete all stored preferences for all users. Use `-prefs user@domain` to delete preferences for the user specified by *user@domain*. Use `-delete user@domain` to delete a user. If the user does not exist in the VCS One cluster but has stored preferences, the user's stored preferences will be deleted.

`-move [-updateroles] [-usergroup] user@domain(s) -ou ouvaluepath [-user user@domain -domaintype domaintype]`

Move a VCS One user or users to the OUValuePath location specified by *ouvaluepath*. Use the `-updateroles` option to update the roles to reflect the change. Use `-usergroup` to move a usergroup.

`-enable [-usergroup] user@domain [-user user@domain -domaintype domaintype]`

Enable a previously disabled user or usergroup, restoring privileges. Use *user@domain(s)* to specify either a user or usergroup.

`-disable [-usergroup] user@domain [-user user@domain -domaintype domaintype]`

Disable a user, removing privileges. Disabled users have no privileges at all. You may also disable a usergroup, using the `-usergroup` option. (Use *user@domain* to specify the usergroup name when disabling a usergroup.)

`-addrole [-usergroup] user@domain role_name [-usergroup] {object(s) | -ou ouvaluepath} [-user user@domain -domaintype domaintype]`

Add a role name (*role_name*) to the user (*user@domain*), and specify the objects (separated by spaces) or the OUValuePath (*ouvaluepath*) for which the role applies.

The objects specified must be of the type indicated by the role type. For example, if *role_name* indicates role type "Group," the objects must be the names of specific service groups. You may also assign a *role_name* to a usergroup, using the `-usergroup` option. (Use *user@domain* to specify the usergroup name when assigning a *role_name* to a usergroup.)

See Examples.

If the user is assigned a role and that user already has equal or greater privileges, the command succeeds with a notification about the user's previously existing roles.

`-deleterole [-usergroup] user@domain role_name [-usergroup] {object(s) | -ou ouvaluepath} [-user user@domain -domaintype domaintype]`

Delete a role (*role_name*) assigned to a user (*user@domain*) for objects for the OUValuePath (*ouvaluepath*).

Specify multiple objects separated by spaces. You may also delete a role assigned to usergroup using the `-usergroup` option. (Use *user@domain* to specify the usergroup name when deleting a role assigned to a usergroup.)

`-display [-sys | -usergroup] [user@domain(s) | -ou ouvaluepath] [-attribute attribute(s)] [-user user@domain -domaintype domaintype]`

Display attribute and value information for a specified user, multiple users, a usergroup, or multiple usergroups. The `-sys` option displays the system (vcsoneclientd) users.

`-display -prefs [user@domain(s) | -all | -ou ouvaluepath] [-attribute attribute(s)] [-user user@domain -domaintype domaintype]`

Display preferences information for a specified user, multiple users, or all users.

`-value [-sys | -usergroup | -prefs] user@domain attribute [-user user@domain -domaintype domaintype]`

Display the value of a specified attribute for a specified user or users. Use the `-sys` option to indicate that the user is a system user. Use the `-usergroup` option to indicate that the *user@domain* that is specified is a usergroup. Use the `-prefs` option to indicate that the *user@domain* that is specified is a user preference.

`-list [-sys | -usergroup] [conditional(s)][-user user@domain -domaintype domaintype]`

List VCS One users. The `-sys` option displays the system users. The `-usergroup` option displays the users in the usergroup.

Use a conditional statement to limit the list. Conditional statements take the form: Attribute=value (equal to), Attribute!=value (greater than), and Attribute=~value (contains). Multiple conditional statements imply AND logic.

`hauser -list -prefs [-all] [conditional(s)] [-user user@domain -domaintype domaintype]`

List preferences information. The `-all` option displays preferences for all users.

Use a conditional statement to limit the list. Conditional statements take the form: Attribute=value (equal to), Attribute!=value (greater than), and Attribute=~value (contains). Multiple conditional statements imply AND logic.

`-modify modify_options`

Modify a user's attributes.

You may modify a scalar attribute's existing value.

Refer to the following list of permissible `-modify` commands.

`SCALAR`

`hauser -modify [-usergroup] user@domain attribute value [-user user@domain -domaintype domaintype]`

Modify a user's attribute value. Use `-usergroup` to modify the attribute value for a usergroup.

`-help [-modify | -list]`

Display the usage information for the command. Use the `-modify` and `-list` options to show usage for these command options. When you enter the command and an option without arguments, syntax for the specific option displays.

`-version`

Display the version information for the `hauser` command.

EXAMPLES

To display help for a specific option, for example, for the `-add` option:

```
# hauser -add
```

To add a user John@abc.com.

```
# hauser -add John@abc.com
```

For user John@abc.com, assign a role named GroupOperator, with operator privileges for service groups A3, A5, and A7.

```
# hauser -addrole John@abc.com GroupOperator A3 A5 A7
```

For the usergroup Operators@abc.com, assign a role named GroupOperator with operator privileges for service groups A3, A5, A7.

```
# hauser -addrole -usergroup Operators@abc.com GroupOperator A3 A5
A7
```

For the user John@abc.com, assign a role named UserOperator with operator privileges for the usergroup HR@abc.com.

```
# hauser -addrole John@abc.com UserOperator -usergroup HR@abc.com
```

For the usergroup Directors@abc.com, assign a role named UserOperator with operator privileges for the usergroup HR@abc.com.

```
# hauser -addrole -usergroup Directors@abc.com UserOperator -usergroup
HR@abc.com
```

To list VCS One users.

```
# hauser -list
```

NOTES

When using the command to specify or modify an attribute's value that begins with a dash ("-"), precede the value with a percent sign ("%"). For example, specify `-y` as `%-y`. Likewise, precede a value that starts with a percent sign with another percent sign.

SEE ALSO

`harole`(1M), `halogin`(1M)

havtype

`havtype` – add, modify, delete, display, and list information about a vtype

SYNOPSIS

```
UNIX: /opt/VRTSvcsone/bin/havtype

Windows: %VCSONE_HOME%\bin\havtype

havtype -display [vtype(s)] [-attribute attribute(s)] [-user
user@domain -domaintype domaintype]
havtype -list [conditional(s)] [-user user@domain -domaintype
domaintype]
havtype -value vtype attribute [-user user@domain -domaintype
domaintype]
havtype -modify modify_options
havtype [-help [-modify|-list]]
havtype -version
```

AVAILABILITY

VRTSvcsonec

DESCRIPTION

A vtype is a virtualization object-type definition that represents a single entity in a virtualization environment. For example, a virtual machine in a VMware ESX environment may be represented by a vtype definition such as "ESXVM". A frame is an object instance of a given vtype. A frame inherits the attributes and defaults of its vtype. A frame's vtype is analogous to a resource's type. Together, frames and vtypes define the hierarchy of any virtualization environment. Vtypes give VCS One a way to understand how to manage individual frames. Their management depends entirely on the characteristics of their vtype.

A vtype defines a set of attributes and defaults that are inherited by a frame object with that vtype.

Use the `havtype` command to display and list information about vtypes. You can also use it to display the attribute value for a given vtype.

A non-root user who has not run the `halogin` command can execute the `havtype` command using the `-user` *user@domain* option to execute the command with the privileges of the specified user. When issuing the command, the user must enter

the fully qualified domain user name and supply a password when prompted. If necessary, the `-domaintype` option can specify the type of domain against which the user is to be authenticated. Supported domain types include:

- "unixpwd"
- "nt"
- "nis"
- "nisplus"
- "ldap"
- "pam"
- "vx" (Symantec Private Domain)

The domain type, by default, is "vx". The domain type is case sensitive.

See NOTES for how to specify "-" and "%" characters in the command line.

OPTIONS

`-display [`*`vtype(s)`*`] [-attribute `*`attribute(s)`*`] [-user `*`user@domain`*
`-domaintype `*`domaintype`*`]`

Displays a vtype(s) or all vtypes if none are specified. Attributes are displayed if specified.

`-list [`*`conditional(s)`*`] [-user `*`user@domain`*` -domaintype `*`domaintype`*`]`

Displays a list of the vtype(s) whose values match the given conditional statements. Conditional statements can take three forms: Attribute=Value, Attribute!=Value, Attribute=~Value. Multiple conditional statements imply AND logic. If no conditional statement is specified, all vtypes in the cluster are listed.

`-value `*`vtype attribute`*` [-user `*`user@domain`*` -domaintype `*`domaintype`*`]`

Provides the value of a single vtype attribute. The `-value` option is used instead of the `-display` option when one specific attribute value is needed rather than a table of many attribute values. Displays the attribute value for the specified vtype.

`-modify `*`modify_options`*

The `-modify` option lets you modify a vtype's attributes.

You may modify a scalar attribute's existing value.

You may not use `-modify` to change values already defined for a vector, a keylist, or an association attribute. For vector, keylist, and association

attributes, the *modify_options*, which include `-add`, `-delete`, `-update`, or `-delete -keys`, may be used.

Refer to the following list of `-modify` commands. You may display the commands using `havtype -help -modify`.

SCALAR

```
havtype -modify vtype attribute value
```

VECTOR

Use the following command only when the attribute has no value:

```
havtype -modify vtype attribute value... [-user user@domain
-domaintype domaintype]
```

For vector attributes that have defined values, only the following operations are allowed:

```
havtype -modify vtype attribute -add value... [-user
user@domain -domaintype domaintype]
```

```
havtype -modify vtype attribute -delete -keys [-user
user@domain -domaintype domaintype]
```

Note: You cannot delete an individual element of a VECTOR.

KEYLIST

Use the following command only when the attribute has no value:

```
havtype -modify vtype attribute {key}... [-user user@domain
-domaintype domaintype]
```

For keylist attributes that have values defined, only the following operations are allowed.

```
havtype -modify vtype attribute -add {key}... [-user
user@domain -domaintype domaintype]
```

```
havtype -modify vtype attribute -delete key... [-user
user@domain -domaintype domaintype]
```

```
havtype -modify vtype attribute -delete -keys [-user
user@domain -domaintype domaintype]
```

ASSOCIATION

Use the following command only when the attribute has no value:

```
havtype -modify vtype attribute {key value}... [-user
user@domain -domaintype domaintype]
```

For association attributes that have values defined, only the following operations are allowed.

```
havtype -modify vtype attribute -add {key value} [-user
user@domain -domaintype domaintype]

havtype -modify vtype attribute -update {key value}... [-user
user@domain -domaintype domaintype]

havtype -modify vtype attribute -delete key... [-user
user@domain -domaintype domaintype]

havtype -modify vtype attribute -delete -keys [-user
user@domain -domaintype domaintype]
```

`[-help [-modify|-list]]`

Displays the command usage for `havtype`. The `-modify` option provides the usage for the `-modify` option and the `-list` option provides the usage for the `-list` option. When you enter `haframe -help` and an option without arguments, the syntax for the specified option displays.

`-version`

Displays the command version.

EXAMPLES

To display the usage syntax for a specific command option, enter the command and an option without arguments. For example, enter:

```
# havtype -value
```

NOTES

When using the command to specify or modify an attribute's value that begins with a dash ("-"), precede the value with a percent sign ("%"). For example, specify `-y` as `%-y`. Likewise, precede a value that starts with a percent sign with another percent sign.

SEE ALSO

`haframe`(1M), `hares`(1M),`hatype`(1M), `haattr`(1M)

vxfentsthdw

`vxfentsthdw` – test SCSI-3 persistent reservations on a disk

SYNOPSIS

`vxfentsthdw [-n][-r[-t|-d|[-m]|[-f` *`filename`*`]|[-g` *`diskgroup`*`]]|-c` *`diskgroup`*`]`

AVAILABILITY

VRTSvcsonecd

DESCRIPTION

The `vxfentsthdw` utility is provided to test disks for support of SCSI-3 persistent reservations. It verifies that the shared storage intended for use can support I/O fencing. The utility works on any two VCS One cluster systems that share disks.

It issues a series of `vxfenadm` commands to set up SCSI-3 registrations on the disk, verifies the registrations on the disk, and removes the registrations from the disk. Note that the utility destroys data on the disks unless the `-r` option is used. The `vxfentsthdw` utility requires that a disk intended for use as a data disk have at least a 10 megabyte capacity.

The `-c` option is not applicable for testing disks used by VCS One client daemon nodes.

This command is not used on Windows.

OPTIONS

`-n`

Use for communications between systems connected to the disk. This option is relevant only for Linux systems, where the communications are SSH by default.

`-r`

Non-destructive testing. Testing of the disks for SCSI-3 persistent reservations occurs in a nondestructive way; that is, there is only testing for reads, not writes. May be used with the `-m`,`-f`, or `-g` options.

`-t`

Testing of the return value of the SCSI TEST UNIT (TUR) command under SCSI-3 reservations. A warning is printed on failure of TUR testing. May be used with the `-m`, `-f`, or `-g` options.

`-d`

Use for devices for which Dynamic Multipathing (DMP) is configured.

`-m`

Manual testing. This is the default option; that is, if no options are specified, the utility carries out the test suite in manual operation. The utility prompts for system names and device paths.

`-f`

Test the disks listed in *filename*. This is a batch test operation. All disks specified in the file are tested one by one. The format of the file is:

Node1Name DevicePath Node2Name DevicePath

EXAMPLES: (Note that the format of DevicePath varies by operating system.)

For `Solaris`, if node SYSA and node SYSB have two shared disks, and the disks are seen as having DevicePath /dev/rdsk/c2t2d1s2 and /dev/rdsk/c3t2d1s2 on SYSA, and /dev/rdsk/c3t2d1s2 and /dev/rdsk/c3t2d2s2 on SYSB, the file *filename* contains:

SYSA /dev/rdsk/c2t2d1s2 SYSB /dev/rdsk/c3t2d1s2

SYSA /dev/rdsk/c3t2d1s2 SYSB /dev/rdsk/c3t2d2s2

For `AIX`, if node SYSA and node SYSB have two shared disks, and the disks are seen as having DevicePath /dev/rhdisk70 and /dev/rhdisk75 on SYSA, and /dev/rhdisk60 and /dev/rhdisk65 on SYSB, the file *filename* contains:

SYSA /dev/rhdisk70 SYSB /dev/rhdisk60

SYSA /dev/rhdisk75 SYSB /dev/rhdisk65

For `Linux`, if node SYSA and node SYSB have two shared disks, and the disks are seen as having DevicePath /dev/sdw and /dev/sdx on SYSA, and /dev/sdy and /dev/sdz on SYSB, the file *filename* contains:

SYSA /dev/sdw SYSB /dev/sdy

SYSA /dev/sdx SYSB /dev/sdz

`-g` *diskgroup*

Test all disks in the *diskgroup*. This option requires that Veritas Volume Manager is installed and running. A test disk group needs to be set up, with all disks to be tested contained within that group. Dynamic Multipathing

(DMP) is tested with this option; that is, the disks contained in the test disk group configured with DMP are tested for SCSI-3 compatibility.

`-c` *`diskgroup`*

The `-c` option is not applicable for testing disks used by VCS One client daemon nodes.

Appendix B

Modifying attribute values from the command line

This appendix includes the following topics:

Introduction

You can modify the values that are assigned to VCS One object attributes. Commands, such as `haclus`, `hagrp`, `hares`, `harole`, `hasys`, `hatype`, or `hauser`, have a `-modify` option.

For example, to change the `Enabled` attribute of a service group from 0 to 1, use a command that resembles:

```
hagrp -modify N-group Enabled 1
```

You can only change an existing VCS One attribute value when the attribute's dimension is *scalar*. A scalar attribute can have only one value. If you try to change an existing value another attribute type, such as an association attribute, VCS One reports an error. VCS One does not allow that type of change to prevent users from overwriting attribute values inadvertently.

This appendix describes how to modify attributes of all dimensions from the command line, using examples and suggestions for using the `-modify` option.

Displaying attribute values

You can use various commands to display the value of an attribute.

To display values for resource types and resources

- Use the `hattr -display` command to display the current values of attributes for a type and the default values of attributes for its resources. For example, to display the attributes of the FileOnOff resource type:

  ```
  hattr -display FileOnOff | more
  ```

Table B-1

#Attribute	DataType	Dimension	Value
ActionTimeout	integer	scalar	30
AgentClass	string	scalar	TS
AgentDirectory	string	scalar	
AgentFailedOn	string	keylist	
AgentFile	string	scalar	
AgentPriority	string	scalar	0
AgentReplyTimeout	integer	scalar	130
AgentStartTimeout	integer	scalar	60
AgentOnlineOps	string	keylist	
ArgList	string	vector	PathName
AttrChangedTimeout	integer	scalar	60
CleanRetryLimit	integer	scalar	0
CleanTimeout	integer	scalar	60
CloseTimeout	integer	scalar	60
ConfInterval	integer	scalar	600
ContainerOpts	integer	assoc	

Table B-1 *(continued)*

Created	integer	scalar	1213317918
FaultOnMonitorTimeouts	integer	scalar	4
FireDrill	boolean	scalar	0
InfoInterval	integer	scalar	0
InfoTimeout	integer	scalar	30
LastConfigUpdate	integer	scalar	0

The display shows the attribute by its name, datatype, dimension, and value(s).

To display values for object level attributes

- To display the attributes and their default values for VCS One cluster, group, system, role, and user objects, use the `haattr -display object` command.

To display values for object level attributes

- To display the attributes and their default values for VCS One cluster, group, system, role, and user objects, use the `haattr -display object` command. For example, to list VCS One group attributes, enter:

```
hattr -display group | more

Attribute defaults for type group

#Attribute                DataType          Dimension  Value
AutoEnableWait            boolean           scalar     0
CompatibleGroups          string            keylist
ContainerInfo             string            assoc
Created                   integer           scalar     0
Enabled                   boolean           scalar     1
Evacuate                  boolean           scalar     1
Frozen                    boolean           scalar     0
GrpFaultPolicy            string            scalar     Failover
IncompatibleGroups        string            keylist
LastConfigUpdate          integer           scalar     0
```

```
LastStateUpdate             integer           scalar     0

Load                        integer           assoc
```

As an alternative on UNIX systems, use the `grep` option to list the value of a specific attribute:

```
haattr -display group | grep Priority

Priority         integer          scalar           5
```

To display values for attributes of specific objects

- You can display the current values of specific attributes for a VCS One cluster, group, system, or user by using the `haxxx -display` *`object`* command. For example, where `ApacheWeb` is the name of a service group, you can enter:

  ```
  hagrp -display ApacheWeb | more

  #Group          Attribute       System          Value

  ApacheWeb       Authority       global          0

  :

  ApacheWeb       SystemList      global          sysA 1 sysB 2
                                                  sysC 3

  :
  ```

  You can display a specific attribute using the `-attribute` option and specifying the specific attribute. For example, to display the SystemList attribute, enter:

  ```
  hagrp -display ApacheWeb -attribute SystemList

  #Group          Attribute       System          Value

  ApacheWeb       SystemList      global          sysA 1 sysB 2
                                                  sysC 3
  ```

Modifying scalar attributes

Scalar attributes have only one value. That value may be an integer or a string. For example, the `Priority` attribute of a system can have a value of 4. You can

change the existing value of a scalar attribute from the command line using the typical syntax:

```
haxxx -modify object attribute value
```

In the syntax, `haxxx` represents the command. For example, the command can be `haclus`, `hagrp`, `hares`, `harole`, `hasys`, `hatype`, or `hauser`. Use a command with the `object` that the attribute and its value apply to. For example, the object and the attribute can apply to a VCS One cluster, service group, resource, system, resource type, or user.

The syntax for each object type is:

```
haclus -modify attribute value

hagrp -modify [-propagate] group attribute value [-sys system]

hares -modify resource attribute value [-sys system]

harole -modify rolename attribute value

hasys -modify [-refreshvars] sys attribute value

hatype -modify type [-platform platform] attribute value

hauser -modify [-usergroup] user@domain attribute value
```

Use the `-sys system` option with `hagrp` and `hares` if you want to modify a localized attribute's value. Use the `-propagate` option with `hagrp` to apply the change to the entire group dependency tree.

To modify a scalar attribute's value: examples

- Use the `hatype` command to change the value of the scalar static attribute:

  ```
  hatype -modify FileOnOff ActionTimeout 50
  ```

  Modify the value of the `Priority` attribute for the group, grpA, from 4 to 3 using hagrp:

  ```
  hagrp -modify grpA Priority3
  ```

Modifying vector attributes

Vector attributes have an *ordered* set of non-unique integer or string values. For example, the MyVector attribute can have an ordered set of integer values 1, 3, 5, and 3.

When modifying a vector attribute, you can take the following actions:

- Use the `-modify` option to assign values to an attribute with no current values.

- Use the `-modify -add` options to add a value to the existing set of values.
- Use the `-delete -keys` options to delete all the existing values. You can then create a new ordered list using the `-modify` option.

Restrictions for modifying vector attributes of VCS One objects include:

- You cannot use the `-modify` option directly to change the existing values of a vector attribute. You must include the `-add` or `-delete -keys`.
- You cannot delete an individual element from an existing set of the ordered values of a vector attribute.

Use the `-sys` *`system`* option with `hagrp` and `hares` if you want to modify a localized attribute's value. Use the `-propagate` option with `hagrp` to apply the change to the entire group dependency tree.

To add initial values to a vector attribute

- Use one of the following commands, depending on the object the attribute applies to. The command fails if the attribute currently has values.

```
haclus -modify attribute value ...

hagrp -modify [-propagate] group attribute value ... [-sys system]

hares -modify resource attribute value ... [-sys system]

hasys -modify system attribute value ...

hatype -modify type attribute value ... [-platform platform]
```

  Remember the list of values you add for a vector attribute is ordered. To create an ordered list of disks for the `MyDisks` resource type attribute, `MyDiskList`, enter the command:

```
hares -modify MyDisks MyDiskList disk1 disk2 disk4
```

To add values to a vector attribute

- Use one of the following commands, depending on which object the attribute applies to.

  ```
  haclus -modify attribute -add key
  ```

  ```
  hagrp -modify [-propagate] group attribute -add key ... [-sys system]
  ```

  ```
  hares -modify resource attribute -add value ... [-sys system]
  ```

  ```
  hasys -modify system attribute -add value ...
  ```

  ```
  hatype -modify type attribute -add value ... [-platform platform]
  ```

  For a resource, `MyDisks` that lists its disks in a specific order, and a resource type attribute, `MyDiskList`, enter the command:

  ```
  hares -modify MyDisks MyDiskList -add disk3
  ```

To delete current values of a vector attribute

- Use one of the following commands, depending on the object the attribute applies to.

  ```
  haclus -modify attribute -delete -keys
  ```

  ```
  hagrp -modify [-propagate] group attribute -delete -keys [-sys system]
  ```

  ```
  hares -modify resource attribute -delete -keys [-sys system]
  ```

  ```
  hasys -modify system attribute -delete -keys
  ```

  ```
  hatype -modify type attribute -delete -keys [-plaform platform]
  ```

  Suppose you want delete all values currently assigned for an attribute:

  ```
  hares -modify MyDisks MyDiskList -delete -keys
  ```

Modifying keylist attributes

Keylist attributes have a set of unique integer or string values, that is, keys, which do not need to be ordered. For example, the keylist attribute may have the values: `Value2 Value4 Value3`.

You cannot use the `-modify` option directly to change the existing values of a keylist attribute. You must include the `-add` or `-delete -keys`.

When modifying a keylist attribute, you can take the following actions:

- Use the `-modify` option to assign values to an attribute with no current values.

- Use the `-modify -add` options to add a value to the existing values.
- Use the `-modify -delete` *key* command to delete an individual attribute's value.
- Use the `-delete -keys` options to delete all the existing values. You can then create a new ordered list using the `-modify` option.

Use the `-sys system` option with `hagrp` and `hares` if you want to modify a localized attribute's value. Use the `-propagate` option with `hagrp` to apply the change to the entire group dependency tree.

To add initial values to a keylist attribute

- Use one of the following commands, depending on the object the attribute applies to. The command fails if the attribute currently has values.

  ```
  haclus -modify attribute key ...

  hagrp -modify [-propagate] group attribute key ... [-sys system]

  hares -modify resource attribute key ... [-sys system]

  hasys -modify system attribute key ...

  hatype -modify type attribute key ... [-platform platform]
  ```

  For example, to change the value of a static attribute of a resource type, use a command resembling:

  ```
  hatype -modify FileOnOff MyStrKeylist Value1
  ```

To add values to a keylist attribute

- Use one of the following commands, depending on the object the attribute applies to.

  ```
  haclus -modify attribute -add key

  hagrp -modify [-propagate] group attribute -add key ...
  [-syssystem]

  hares -modify resource attribute -add key ... [-sys system]

  hasys -modify system attribute -add key ...

  hatype -modify type attribute -add key ... [-platform platform]
  ```

  For example, to add values to a keylist attribute:

  ```
  hagrp -modify GrpA MyList -add Value2 Value3
  ```

To delete a keylist attribute value

- You can delete a value of a keylist attribute. Use one of the following commands, depending on the object the attribute applies to.

  ```
  haclus -modify attribute -delete key
  ```

  ```
  hagrp -modify [-propagate] group attribute -delete key ... [-sys system]
  ```

  ```
  hares -modify resource attribute -delete key ... [-sys system]
  ```

  ```
  hasys -modify system attribute -delete key ...
  ```

  ```
  hatype -modify type attribute -delete key ... [-platform platform]
  ```

  For example, to delete a value for a keylist attribute:

  ```
  hagrp -modify GrpA MyGrpKeyListAttr -delete Value3
  ```

To delete all current keylist values

- Use one of the following commands, depending on the object the attribute applies to.

  ```
  haclus -modify attribute -delete -keys
  ```

  ```
  hagrp -modify [-propagate] group attribute -delete -keys [-sysystem]
  ```

  ```
  hares -modify resource attribute -delete -keys [-sys system]
  ```

  ```
  hasys -modify system attribute -delete -keys
  ```

  ```
  hatype -modify type attribute -delete -keys [-plaform platform]
  ```

  For example, to delete all values from a service group's keylist attribute:

  ```
  hagrp -modify grpB DiskList -delete -keys
  ```

Modifying association attributes

Association attributes have a set of unordered key-value pairs, which may have integer or string values. For example, an attribute may have the values: `AssocKey1 10 AssocKey3 13 AssocKey2 11`.

You cannot use the `-modify` option directly to change the existing values of a keylist attribute. You must include the `-add`, `-update`, `-delete`, or `-delete -keys`.

When modifying a keylist attribute, you can take the following actions:

- Use the `-modify` option to assign values to an attribute with no current key-value pairs.

- Use the `-modify -add` options to add a key-value pair to an attribute's existing key-value pairs.
- Use the `-modify -update` options to update the value of a key-value pair. The existing values are replaced with the new values you specify.
- Use the `-modify -delete key` command to delete a key-value pair of an individual attribute.
- Use the `-delete -keys` options to delete all the existing key-value pairs. You can then create a new ordered list using the `-modify` option.

Use the `-sys system` option with `hagrp` and `hares` if you want to modify a localized attribute's value. Use the `-propagate` option with `hagrp` to apply the change to the entire group dependency tree.

To add initial key-value pairs for an association attribute

- Use one of the following commands, depending on the object the attribute applies to. The command fails if the attribute currently has values. See the next sections.

  ```
  haclus -modify attribute {key value} ...
  ```

  ```
  hagrp -modify [-propagate] group attribute {key value}... [-sys system]
  ```

  ```
  hares -modify resource attribute {key value} ... [-sys system]
  ```

  ```
  hasys -modify system attribute {key value} ...
  ```

  ```
  hatype -modify type attribute {key value}...[-platform platform]
  ```

  For example, to add key-value pairs for a static attribute of a resource type that currently has no key-values, use a command resembling:

  ```
  hatype -modify FileOnOff MyAssoc Key1 1 Key2 2 -platform linux
  ```

To add key-value pairs to an existing association attribute

- You can add a key-value pair to an association type attribute that already has key-value pairs. Use one of the following commands, depending on the object the attribute applies to.

  ```
  haclus -modify attribute -add {key value} ...
  ```

  ```
  hagrp -modify [-propagate] group attribute -add {key value} ...
  [-sys system]
  ```

  ```
  hares -modify resource attribute -add {key value} ...
  ```

  ```
  hasys -modify system attribute -add {key value} ...
  ```

  ```
  hatype -modify -add {key value} ... [-platform platform]
  ```

  For example, to add the key-value pair, `MyIntKey11`, to the `MyAssocAttr` resource, `MyResource`, use the command:

  ```
  hares -modify -add MyResource MyAssocAttr MyIntKey1 1
  ```

To update existing association attribute key-value pairs

- You can update values of existing key-value pairs of an association attribute. Use one of the following commands, depending on the object the attribute applies to.

  ```
  haclus -modify attribute -update {key value} ...
  ```

  ```
  hagrp -modify [-propagate] group attribute -update {key value}
  ... [-sys system]
  ```

  ```
  hares -modify resource attribute -update {key value} ... [-sys
  system]
  ```

  ```
  hasys -modify system attribute -update {key value} ...
  ```

  ```
  hatype -modify -update type attribute {key value} ... [-platform
  platform]
  ```

  In the following example, the command changes the key-value, `GrpKey 1` to `GrpKey 2` for the attribute `MyAssocAttr`:

  ```
  hagrp -modify MyGroup MyAssocAttr -update GrpKey1 2
  ```

To delete an existing association attribute key

- You can delete an existing key-value pair of an association attribute. Use one of the following commands, depending on the object the attribute applies to.

  ```
  haclus -modify attribute -delete key ...
  ```

  ```
  hagrp -modify [-propagate] group attribute -delete key ... [-sys system]
  ```

  ```
  hares -modify resource attribute -delete key ... [-sys system]
  ```

  ```
  hasys -modify system attribute -delete key ...
  ```

  ```
  hatype -modify type attribute -delete key ... [-platform platform]
  ```

  In the following example, the command removes the key-value pair `Key1 2`from the group attribute `MyGrpAttr`:

  ```
  hagrp -modify MyGrp MyGrpAttr -delete Key1 2
  ```

To delete an association attribute's existing keys

- You can delete all existing key-value pairs of an association attribute. Use one of the following commands, depending on the object the attribute applies to.

  ```
  haclus -modify attribute -delete -keys ...
  ```

  ```
  hagrp -modify [-propagate] group attribute -delete -keys ... [sys system]
  ```

  ```
  hares -modify resource attribute -delete -keys ... [-sys system]
  ```

  ```
  hasys -modify system attribute -delete -keys ...
  ```

  ```
  hatype -modify type attribute -delete -keys ... [-platform platform]
  ```

  For example, delete all key-value pairs of the association attribute `MySysAttr` from the system `SysA`:

  ```
  hasys -modify SysA MySysAttr -delete -keys
  ```