Veritas Access Command Reference Guide Documentation

Release 7.3.1

RHEL 6.x

Linux

Dec 13, 2017

Contents

1 Legal Notice

Last updated: Dec 13, 2017

Veritas Technologies LLC
500 E Middlefield Road
Mountain View, CA 94043
http://www.veritas.com

1.1 Technical Support

Technical Support maintains support centers globally. All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policies. For information about our support offerings and how to contact Technical Support, visit our website:

https://www.veritas.com/support

You can manage your Veritas account information at the following URL:

https://my.veritas.com

If you have questions regarding an existing support agreement, please email the support agreement administration team for your region as follows:

Worldwide (except Japan) CustomerCare@veritas.com
Japan Customer Care_Japan@veritas.com

1.2 Documentation

Make sure that you have the current version of the documentation. Each document displays the date of the last update on the cover page. The latest documentation is available on the Veritas website:

https://sort.veritas.com/documents

1.3 Documentation feedback

Your feedback is important to us. Suggest improvements or report errors or omissions to the documentation. Include the document title, document version, chapter title, and section title of the text on which you are reporting. Send feedback to:

doc.feedback@veritas.com

You can also see documentation information or ask a question on the Veritas community site:

http://www.veritas.com/community/

1.4 Veritas Services and Operations Readiness Tools (SORT)

Veritas Services and Operations Readiness Tools (SORT) is a website that provides information and tools to automate and simplify certain time-consuming administrative tasks. Depending on the product, SORT helps you prepare for installations and upgrades, identify risks in your datacenters, and improve operational efficiency. To see what services and tools SORT provides for your product, see the data sheet:

https://sort.veritas.com/data/support/SORT_Data_Sheet.pdf

2 Introduction

2.1 Veritas Access CLI command conventions

This document uses the following conventions when describing commands that are used in the Veritas Access CLI, also sometimes referred to as the CLISH.

2.1.1 Command conventions

- Brackets [] indicate that the enclosed component of the command line is optional.
- Curly braces { } indicate an association between the enclosed options. For example, {opt1 [opt2 ...optn]} means that if the command contains opt1, then the command may optionally contain opt2 ...optn.
- A vertical bar (or the pipe symbol) | separates optional arguments from which you can choose. For example, if a command has the following format, you can choose `arg1` or `arg2` (but not both):

  `command [arg1 | arg2 ]`
- Italics indicate that the information is user supplied. For example, the user supplies the pool name, disk, and if the storage pool is isolated or not in the following command:

  `Storage> pool create` *pool_name disk1* [, *disk2*...] {`isolated=[yes|no]`}
- An ellipsis (...) means that you can repeat the previous parameter. For example, consider the following command:

  `Storage> pool adddisk` *pool_name disk1* [, *disk2*,...]

2.2 About accessing the online man pages

You access the online man pages by typing `man name_of_command` at the command line.

The example shows the result of entering the `History> man` command.

NAME `history` - display command history

SYNOPSIS `history` [*username*] [*number_of_lines*]

`DESCRIPTION` The `history` command displays the commands executed by the user in the following format: <time stamp> <status> <message> <command>

```
<time stamp> : MM-DD-YYYY HH:MM
<status> : Success | Error | Warning
<message> : Description of the command.
<command> : Actual command executed by the user
```

You can also type a question mark (?) at the prompt for a list of all the commands that are available for the command mode that you are in. For example, if you are within the `Admin` mode, if you type a question mark (`?`), you see a list of the available commands for the `Admin` mode.

```
ACCESS> admin ?
Entering admin mode...
ACCESS.Admin>
exit         -- Return to the previous menus
logout       -- Logout of the current CLI session
man          -- Display on-line reference manuals
passwd       -- Change the administrator password
show         -- Show the administrator details
supportuser  -- Enable or disable the support user
user         -- Add or delete an administrator
```

To exit the command mode, enter the following command: `exit` To exit the system console, enter the following command: `logout`.

3 Admin Commands

3.1 admin

3.1.1 SYNOPSIS

`show` [*username*]

`passwd` [*username*]

`supportuser disable`

`supportuser enable`

`supportuser password`

`supportuser status`

`user add` *username* system-admin|storage-admin|master

`user delete` *username*

3.1.2 DESCRIPTION

These `admin` commands are used to display a list of administrators, to add or delete an administrator, and to change the password of an administrator. An administrator can be a `Master`, `System Administrator`, or a `Storage Administrator`. A `Master` Administrator has all the permissions. Only the `Master` Administrator can add or delete other administrators.

3.1.3 OPTIONS

`show` [*username*] Show the administrator details of the given username.

`passwd` [*username*] Change the password of the given username. Only the `Master` Administrator can change the password of other administrators.

`supportuser disable` Disallow non-local logins to `support` user. Only the `Master` Administrator can disable the `support` user. By default, `support` user is enabled after installation of Veritas Access.

supportuser enable Allow `support` user to login from anywhere. Only the `Master` Administrator can enable the `support` user.

supportuser password Change the password of the `support` user. Only the `Master` Administrator can change the password of the `support` user.

supportuser status Show the status (either enable or disable) of the `support` user. Only the `Master` Administrator can see the status of the `support` user.

user add ***username*** **system-admin|storage-admin|master** Add a new administrator with the given username and privileges.

user delete ***username*** Delete an administrator with the given username.

3.1.4 EXAMPLES

Display all the administrators on the current system.

```
Admin> show
List of Users
-------------
master
```

Display the details of the administrator with username *master*.

```
Admin> show master
Username      : master
Privileges    : Master
```

Disable the *support* user.

```
Admin> supportuser disable
Disabling support user.
support user disabled.
```

Enable the *support* user.

```
Admin> supportuser enable
Enabling support user.
support user enabled.
```

Display the status of the *support* user.

```
Admin> supportuser status
support user status : Enabled
```

3.1.5 SEE ALSO

user(1), passwd(1), show(1)

3.2 passwd

3.2.1 SYNOPSIS

`passwd` [*username*]

3.2.2 DESCRIPTION

The admin `passwd` command is used to change the user's password. Only the `Master` administrator can change the password of the other administrators. Note: The system does not ask for the old password.

3.2.3 OPTIONS

username Change the password of the specified user. If *username* is not specified, the password of the currently logged-in user is changed.

3.2.4 EXAMPLES

Change the password of the currently logged-in user.

```
Admin> passwd
```

Change the password of the `master` user.

```
Admin> passwd master
```

3.2.5 SEE ALSO

show(1), user(1)

3.3 show

3.3.1 SYNOPSIS

`show` [*username*]

3.3.2 DESCRIPTION

The admin `show` command displays the user information that includes the user name and privileges.

3.3.3 OPTIONS

username Show details of the specified user. If the *username* is not specified, the command displays a list of all of the users.

3.3.4 EXAMPLES

Display all the administrators on the current system.

```
Admin> show
List of Users
-------------
master
```

Display the details of the administrator with username *master*.

```
Admin> show master
Username      : master
Privileges    : Master
```

3.3.5 SEE ALSO

user(1), passwd(1)

3.4 supportuser

3.4.1 SYNOPSIS

`supportuser disable`

`supportuser enable`

`supportuser password`

`supportuser status`

3.4.2 DESCRIPTION

The `supportuser` commands disable, enable, and show the status of the `support` user. The commands are also used to change the password of the `support` user.

3.4.3 OPTIONS

`supportuser disable` Disable remote logins to `support` user. Only the `Master` Administrator can disable the `support` user. By default, the `support` user is enabled after installation of Veritas Access.

`supportuser enable` Allow the `support` user to login from anywhere. Only the `Master` Administrator can enable the `support` user.

`supportuser password` Change the password of the `support` user. Only the `Master` Administrator can change the password of the `support` user.

`supportuser status` Show the status (either enable or disable) of the `support` user. Only the `Master` Administrator can see the status of the `support` user.

3.4.4 EXAMPLES

Disable the *support* user.

```
Admin> supportuser disable
Disabling support user.
support user disabled.
```

Enable the *support* user.

```
Admin> supportuser enable
Enabling support user.
support user enabled.
```

Display the status of the *support* user.

```
Admin> supportuser status
support user status : Enabled
```

3.4.5 SEE ALSO

user(1), passwd(1), show(1)

3.5 user

3.5.1 SYNOPSIS

`user add` *username* system-admin|storage-admin|master

`user delete` *username*

3.5.2 DESCRIPTION

The admin `user` commands add or delete a user. A user can be a `Master` user who has all of the permissions, including adding and deleting users. By default, the password of the new user is the same as the username. A Storage Administrator has access to only storage commands and is responsible for upgrading the cluster and applying the patches. A System Administrator is responsible for configuring the NFS server and exporting the file system, adding or deleting new nodes to the cluster, and configuring other network parameters like DNS, NIS, and so on.

3.5.3 OPTIONS

`user add` *username* system-admin|storage-admin|master Add a user with specific privileges.

`user delete` *username* Delete the specified user from the current system.

3.5.4 EXAMPLES

Add a `Master` with username `master1`.

```
Admin> user add master1 Master
```

Add a `System Administrator` with username `admin`.

```
Admin> user add admin System-admin
```

Delete a user with username `admin`.

```
Admin> user delete admin
```

3.5.5 SEE ALSO

show(1)

4 Backup Commands

4.1 backup

4.1.1 SYNOPSIS

`netbackup master-server` *server*

`netbackup emm-server` *server*

`netbackup media-server add` *server*

`netbackup media-server delete` *server*

`netbackup sanclient enable|disable`

`netbackup exclude_list show` [*policy*] [*schedule*]

`netbackup exclude_list add` *pattern* [*policy*] [*schedule*]

`netbackup exclude_list delete` *pattern* [*policy*] [*schedule*]

`netbackup include_list show` [*policy*] [*schedule*]

`netbackup include_list add` *pattern* [*policy*] [*schedule*]

`netbackup include_list delete` *pattern* [*policy*] [*schedule*]

`install` *version* [*url*]

`uninstall`

`virtual-ip` *ipaddr* [*device*]

`virtual-name` *name*

`show`

`status`

`start` [*nodename*]

`stop`

4.1.2 DESCRIPTION

The backup commands configure the built-in NetBackup client software.

4.1.3 NetBackup Client

Each node in the cluster includes built-in NetBackup client software of version 7.7 for users with the NetBackup data protection suite. The NetBackup Master Server and NetBackup Media Server should be 7.7 or higher.

NetBackup installed on the cluster nodes works as a NetBackup standard client to perform IP-based backups and a NetBackup SAN client to perform SAN-based backups of file systems.

The backup `netbackup` commands configure the local NetBackup installation to work with an external NetBackup Master Server.

The backup `virtual-name` command configures a NetBackup installation on the cluster nodes to use *name* as its hostname.

The backup `netbackup master-server`, `netbackup emm-server`, and `netbackup media-server` commands configure the NetBackup client to work with the given NetBackup installation.

The backup `virtual-ip` command configures the built-in NetBackup Client installation to use *ipaddr* as its virtual IP address and *device* as the network interface for the virtual ip.

The backup `install` command installs a one higher or one lower supported version of the Netbackup client.

The backup `uninstall` command uninstalls the currently installed version of the Netbackup Client.

Note: The NetBackup SAN client should only be enabled if the required licenses are installed on the NetBackup Master Server. If the required licenses for the NetBackup SAN client are not available on the NetBackup Master Server, then the backup service fails to start. If you do not have the required license for the NetBackup SAN client, then you must disable the SAN client using `netbackup sanclient disable`. Otherwise, the backup service fails to start.

The built-in NetBackup client software also supports snapshot-based backups. To use snapshot-based backups with the built-in NetBackup client, use snapshot type as `VxFS_Checkpoint` in the NetBackup policy configuration.

4.1.4 OPTIONS

server Hostname of the server to use as the NetBackup Master Server, EMM Server, or Media Server. Make sure that *server* can be resolved through DNS and its IP address can be resolved back to *server* through the DNS reverse lookup.

ipaddr Virtual IP address to be used by the NetBackup installation on Veritas Access nodes. Make sure that *ipaddr* can be resolved back to the hostname configured through the backup `virtual-name` command for proper functioning of NetBackup.

name Hostname to be used by a NetBackup installation on the Veritas Access nodes. Make sure that *name* can be resolved through DNS and its IP address can be resolved back to *name* through a DNS reverse lookup. Also make sure that *name* resolves to the IP address configured through the backup `virtual-ip` command.

version Netbackup client version to be installed. Netbackup versions 7.7 and 7.6 are currently supported.

url URL specifying the location of the Netbackup client to be installed. Currently, supported protocols are http, ftp, and scp.

`virtual-ip` ***ipaddr*** **[*device*]** Configure the NetBackup installation on the Veritas Access nodes to use *ipaddr* as its highly available virtual IP address and *device* as the network interface for the virtual IP address.

To remove the virtual IP address configured for backup, use "-" for *ipaddr*. Like, "virtual-ip -".

A network mask can be provided with the IP address as the prefix length for the *ipaddr*. For example, `virtual-ip 10.209.107.151/22`.

IPv6 addresses can be specified as "virtual-ip fec0::20c:29ff:fe8e:2203/64".

`virtual-name` ***name*** Configure a NetBackup installation on Veritas Access nodes to use *name* as its hostname. Make sure that name resolves to the IP address that is configured using the `virtual-ip` command.

`netbackup master-server` ***server*** Configure the hostname of the NetBackup Master Server name to be used by the built-in NetBackup client installation. This setting is mandatory, if the NetBackup client software needs to be used. Make sure that the *server* hostname can be resolved through DNS, and its IP address can be resolved back to *server* hostname through the DNS reverse lookup.

`netbackup emm-server` ***server*** Configure an external NetBackup EMM Server name to be used by the NetBackup installation on the Veritas Access nodes. This setting is mandatory, if the NetBackup client software needs to be used. Make sure that the *server* hostname can be resolved through DNS, and its IP address can be resolved back to *server* hostname through the DNS reverse lookup.

`netbackup media-server add` ***server*** Add an external NetBackup Media Server name to be used by the NetBackup installation on the Veritas Access nodes. This setting is optional, since if the Media Server name is not added, the local NetBackup client installation uses the NetBackup Master Server as the NetBackup Media Server. Make sure that the *server* hostname can be resolved through DNS, and its IP address can be resolved back to *server* hostname through the DNS reverse lookup.

.nf `netbackup media-server delete` *server* .fi Delete an already configured NetBackup Media Server name.

`netbackup sanclient enable|disable` Enable or disable the NetBackup SAN client. The NetBackup SAN client should only be enabled if the required licenses are installed on the NetBackup Master Server. If the required licenses for the NetBackup SAN client are not available on the NetBackup Master Server, then the backup service fails to start.

`netbackup exclude_list show` **[*policy*] [*schedule*]** Show the entries in the exclude list entries. Exclude entries can also be explicitly listed for a specific policy by using the *policy* attribute. If *schedule* is specified, then the exclude list entries for the policy specified in *policy* and schedule *schedule* are shown.

`netbackup exclude_list add` ***pattern*** **[*policy*] [*schedule*]** Add the given pattern to the list of files to be excluded. File names matching *pattern* are excluded from backups. If *policy*, then the *pattern* is only applicable for backups of the specific policy.

`netbackup exclude_list delete` ***pattern*** **[*policy*] [*schedule*]** Delete the given pattern from the list of patterns to be excluded.

`netbackup include_list show` **[*policy*] [*schedule*]** Show the entries in the include list entries. Include entries can also be explicitly listed for a specific policy by using the *policy* attribute. If *schedule* is specified, then include list entries for the policy specified in *policy* and schedule *schedule* are shown.

`netbackup include_list add` ***pattern*** **[*policy*] [*schedule*]** Add the given pattern to the list of files to be included. File names matching *pattern* are always included in backups. If *policy*, then the *pattern* is only applicable for backups of the specific policy.

netbackup include_list delete ***pattern*** **[*policy*] [*schedule*]** Delete the given pattern from the list of patterns to be included.

install ***version*** **[*url*]** The backup `install` command installs a one higher or one lower supported version of the NetBackup client.

uninstall The backup `uninstall` command uninstalls the currently installed version of the NetBackup client.

show This command shows the configured settings.

status Display the status of the current backup activities.

start [*nodename*] Start processes that handle backup and restore. Also online a virtual IP address.

stop Stop processes that handle backup and restore. Also offline a virtual IP address.

4.1.5 EXAMPLES

Add an external NetBackup Master Server.

```
    Backup> netbackup master-server nbumaster.veritas.com
Success.
```

Add an external NetBack EMM Server name (which can be the same as the NetBackup Master Server).

```
    Backup> netbackup emm-server nbumaster.veritas.com
Success.
```

Add an external NetBackup Media Server name (if the NetBackup Media Server is not co-located with the NetBackup Master Server).

```
    Backup> netbackup media-server add nbumedia.veritas.com
Success.
```

Delete an already added external Media Server name.

```
    Backup> netbackup media-server delete nbumedia.veritas.com
Success.
```

Configure or change the virtual IP address and network interface of the NetBackup client on the Veritas Access nodes.

```
    Backup> virtual-ip 10.10.10.10 pubeth1
Success.
```

Enable the NetBackup SAN client.

```
    Backup> netbackup sanclient enable
Success.
```

Configure or change the hostname of the NetBackup installation on the Veritas Access nodes.

```
    Backup> virtual-name nbuclient.veritas.com
Success.
```

Install Netbackup Client 7.7 using a Netbackup client tar with SCP.

```
    Backup> install 7.7 scp://root@10.209.106.59:/home/NetBackup_7.7.3_CLIENTS2.tar.gz
Copying from scp://root@10.209.106.59:/home/NetBackup_7.7.3_CLIENTS2.tar.gz
Password:
Copying from scp://root@10.209.106.59:/home/NetBackup_7.7.3_CLIENTS2.tar.gz completed
100% [#] NBU Client Installation completed
```

Install NetBackup client 7.7 using a NetBackup client tar with FTP.

```
    Backup> install 7.7 ftp://ftp@10.209.106.59:/NetBackup_7.7.3_CLIENTS2.tar.gz
Copying from ftp://ftp@10.209.106.59:/NetBackup_7.7.3_CLIENTS2.tar.gz
Password:
Copying from ftp://ftp@10.209.106.59:/NetBackup_7.7.3_CLIENTS2.tar.gz completed
100% [#] NBU Client Installation completed
```

Install NetBackup client 7.7 using a NetBackup client tar with HTTP.

```
    Backup> install 7.7 http://root@172.16.0.3:/NetBackup_7.7.3_CLIENTS2.tar.gz
Copying from http://root@172.16.0.3:/NetBackup_7.7.3_CLIENTS2.tar.gz
Enter password for user 'root':
Copying from http://root@172.16.0.3:/NetBackup_7.7.3_CLIENTS2.tar.gz completed
100% [#] NBU Client Installation completed
```

Uninstall the currently installed NetBackup client.

```
    Backup> uninstall
100% [#] NBU Client uninstallation completed
```

Show the configured settings.

```
    Backup> show
Virtual Name:              nbuclient.veritas.com
Virtual IP:                10.10.10.10/24
NetBackup Master Server:   nbumaster.veritas.com
NetBackup EMM Server:      nbumaster.veritas.com
NetBackup Media Server(s): not configured
Bakcup Device:             pubeth1
```

Display the status when no backup jobs are running.

```
    Backup> status

Virtual IP state               : up
Backup service online node     : node_01
NetBackup Client state         : running
NetBackup SAN Client state     : running

No backup/restore jobs running.
```

Display the status when backup jobs are running that involve file systems using the NetBackup client.

```
    Backup> status

Virtual IP state               : up
Backup service online node     : node_01
NetBackup Client state         : working
NetBackup SAN Client state     : running

Some filesystems are busy in backup/restore jobs by NetBackup Client
```

Start processes that handle backup and restore.

```
    Backup> start
NetBackup Client started successfully.
Success.
```

Stop processes that handle backup and restore when backup jobs are running.

```
    Backup> stop
ACCESS backup ERROR V-288-0 Cannot stop, some backup jobs are running.
```

4.1.6 SEE ALSO

netbackup(1), virtual-ip(1), virtual-name(1), show(1), status(1), start(1), stop(1), install(1), uninstall (1)

4.2 install

4.2.1 SYNOPSIS

`install` *version* [*url*]

4.2.2 DESCRIPTION

The backup `install` command installs a one higher or one lower supported version of the NetBackup client. Each node in the cluster includes built-in NetBackup client software of version 8.0 for users with the NetBackup data protection suite.

4.2.3 OPTIONS

version NetBackup client version to be installed. NetBackup versions 8.0 and 7.7 are currently supported.

url URL specifying the location of the NetBackup client to be installed. Currently, supported protocols are http, ftp, and scp.

4.2.4 EXAMPLES

Install the NetBackup client 7.7 using a NetBackup client tar with SCP.

```
Backup> install 7.7 scp://root@10.209.106.59:
/home/NetBackup_7.7.1_CLIENTS2.tar.gz

Copying from scp://root@10.209.106.59:/home/NetBackup_7.7.1_CLIENTS2.tar.gz
Password:
Copying from scp://root@10.209.106.59:/home/NetBackup_7.7.1_CLIENTS2.tar.gz completed
100% [#] NBU Client Installation completed
```

4.2.5 SEE ALSO

backup(1), netbackup(1), virtual-ip(1), virtual-name(1), show(1), status(1), stop(1), install(1), uninstall(1)

4.3 netbackup

4.3.1 SYNOPSIS

`netbackup master-server set` *server*

`netbackup master-server reset`

`netbackup emm-server set` *server*

`netbackup emm-server reset`

`netbackup media-server add` *server*

`netbackup media-server delete` *server*

`netbackup sanclient enable|disable`

`netbackup exclude_list show` [*policy*] [*schedule*]

`netbackup exclude_list add` *pattern* [*policy*] [*schedule*]

`netbackup exclude_list delete` *pattern* [*policy*] [*schedule*]

`netbackup include_list show` [*policy*] [*schedule*]

`netbackup include_list add` *pattern* [*policy*] [*schedule*]

`netbackup include_list delete` *pattern* [*policy*] [*schedule*]

4.3.2 DESCRIPTION

Each node in the cluster includes built-in NetBackup client software of version 7.7 for users with the NetBackup data protection suite. The NetBackup Master server and NetBackup Media Server should be 7.7 or higher. See the fIVeritas Access Installation GuidefP for the supported NetBackup versions.

NetBackup installed on the cluster nodes works as a NetBackup standard client to perform IP-based backups. The NetBackup SAN client performs SAN-based backups of file systems.

The backup `netbackup` commands configure the local NetBackup installation with an external NetBackup Master Server.

The backup `virtual-name` command configures a NetBackup installation on cluster nodes to use *name* as its hostname.

The backup `netbackup master-server`, `netbackup emm-server`, and `netbackup media-server` configures the NetBackup client to work with the given NetBackup installation.

The backup `virtual-ip` command configures the built-in NetBackup client to use *ipaddr* as its virtual IP address and *device* as the network interface for the virtual IP.

Note: The NetBackup SAN client should only be enabled if the required licenses are installed on the NetBackup Master Server. If the required licenses for the NetBackup SAN client are not available on the NetBackup Master Server, then the backup service fails to start. If you do not have the required license for the NetBackup SAN client, then you must disable the SAN client using `netbackup sanclient disable`. Otherwise, the backup service fails to start.

The built-in NetBackup client software also supports snapshot-based backups. To use snapshot-based backups with the built-in NetBackup client, use snapshot type as `VxFS_Checkpoint` in the NetBackup policy configuration.

4.3.3 OPTIONS

server Hostname of the server to use as the NetBackup Master Server, EMM Server, or Media Server. Make sure that *server* can be resolved through DNS and its IP address can be resolved back to *server* through the DNS reverse lookup.

`netbackup master-server set` ***server*** Configure the hostname of the NetBackup Master Server name to be used by the built-in NetBackup client installation. This setting is mandatory if the NetBackup client software needs to be used. Make sure that the *server* hostname can be resolved through DNS, and its IP address can be resolved back to *server* hostname through the DNS reverse lookup.

`netbackup emm-server set` ***server*** Configure an external NetBackup EMM Server name to be used by the NetBackup installation on the Veritas Access nodes. This setting is mandatory, if the NetBackup client software needs to be used. Make sure that the *server* hostname can be resolved through the DNS, and its IP address can be resolved back to *server* hostname through the DNS reverse lookup.

`netbackup media-server add` ***server*** Add an external NetBackup Media Server name to be used by the NetBackup installation on the Veritas Access nodes. This setting is optional, since if the Media Server name is not added, the local NetBackup client installation uses the NetBackup Master Server as the NetBackup Media Server. Make sure that the *server* hostname can be resolved through DNS, and its IP address can be resolved back to *server* hostname through the DNS reverse lookup.

`netbackup media-server delete` ***server*** Delete an already configured NetBackup Media Server name.

`netbackup sanclient enable|disable` Enable or disable the NetBackup SAN client. The NetBackup SAN client should only be enabled if the required licenses are installed on the NetBackup Master Server. If the required licenses for the NetBackup SAN client are not available on the NetBackup Master Server, then the backup service fails to start.

`netbackup exclude_list show` **[*policy*] [*schedule*]** Show the entries in the exclude list entries. Exclude entries can also be explicitly listed for a specific policy by using the *policy* attribute. If *schedule* is specified, then exclude list entries for the policy specified in *policy* and schedule *schedule* are shown.

`netbackup exclude_list add` ***pattern*** **[*policy*] [*schedule*]** Add the given pattern to the list of files to be excluded. File names matching *pattern* are excluded from backups. If *policy* is used, then the *pattern* is only applicable for backups of the specific policy.

`netbackup exclude_list delete` ***pattern*** **[*policy*] [*schedule*]** Delete the given pattern from the list of patterns to be excluded.

`netbackup include_list show` **[*policy*] [*schedule*]** Show tne entries in the include list entries. Include entries can also be explicitly listed for a specific policy by using the *policy* attribute. If *schedule* is specified, then include list entries for the policy specified in *policy* and schedule *schedule* are shown.

`netbackup include_list add` ***pattern*** **[*policy*] [*schedule*]** Add the given pattern to the list of files to be included. File names matching *pattern* are always included in backups. If *policy* is used, then the *pattern* is only applicable for backups of the specific policy.

`netbackup include_list delete` ***pattern*** **[*policy*] [*schedule*]** Delete the given pattern from the list of patterns to be included.

4.3.4 EXAMPLES

Add an external NetBackup Master Server.

```
Backup> netbackup master-server set nbumaster.veritas.com
Success.
```

Add an external NetBackup EMM Server name (which can be the same as the NetBackup Master Server).

```
Backup> netbackup emm-server set nbumaster.veritas.com
Success.
```

Add an external NetBackup Media Server name (if the NetBackup Media Server is not co-located with the NetBackup Master Server).

```
Backup> netbackup media-server add nbumedia.veritas.com
Success.
```

Delete an already added external Media Server name.

```
Backup> netbackup media-server delete nbumedia.veritas.com
Success.
```

Enable the NetBackup SAN client.

```
Backup> netbackup sanclient enable
Success.
```

Show the exclude list entries.

```
Backup> netbackup exclude_list show
Pattern           Policy            Schedule
-------           ------            --------
hosts                 -                 -
iscsid.conf       -                 -
iscsid.conf       NBU_nasgw12       -
/vx/fs100/as*     policy            -
/vx/fs100/*mp3    policy            -
/vx/fs200/bs*     policy2           sched3
```

Add an entry to the include list for policy with name backup_policy3.

```
Backup> netbackup include_list add *.jpg backup_policy3
Success.
```

4.3.5 SEE ALSO

backup(1), virtual-ip(1), virtual-name(1), show(1), status(1), start(1), stop(1)

4.4 show

4.4.1 SYNOPSIS

`show`

4.4.2 DESCRIPTION

The backup `show` command displays the configured settings. The displayed settings may not be currently used by the Veritas Access nodes. When some settings are configured, there might be backup and restore jobs that are running, and so those settings do not immediately take effect. To fill this gap, you have to use the backup `stop` command followed by the backup `start` command.

4.4.3 OPTIONS

`show`

This command shows the configured settings.

4.4.4 EXAMPLES

Show the configured settings.

```
Backup> show
Virtual Name:                    nbuclient.veritas.com
Virtual IP:                      10.10.10.10/24
NetBackup Master Server:         nbumaster.veritas.com
NetBackup EMM Server:            nbumaster.veritas.com
NetBackup Media Server(s):       not configured
Backup Device:                   pubeth1
NetBackup Client Version:        8.0
NetBackup global log level:      not configured
NetBackup database log level:    not configured
Enable robust logging:           not configured
Enable critical process logging: not configured
```

4.4.5 SEE ALSO

backup(1), netbackup(1), virtual-ip(1), virtual-name(1), status(1), start(1), stop(1)

4.5 start

4.5.1 SYNOPSIS

`start` [*nodename*]

4.5.2 DESCRIPTION

The backup `start` command starts backup processes that handle backup and restore. It will do nothing if those processes are currently running. It places the virtual IP address configured through the backup `virtual-ip` command in online mode on any of the currently active nodes or *nodename* if specified.

4.5.3 OPTIONS

`start`

Start processes that handle backup and restore. Also places the virtual IP address in online mode.

4.5.4 EXAMPLES

Start processes that handle backup and restore.

```
Backup> start
NetBackup Client started successfully.
```

4.5.5 SEE ALSO

backup(1), netbackup(1), virtual-ip(1), virtual-name(1), show(1), status(1), stop(1)

4.6 status

4.6.1 SYNOPSIS

`status`

4.6.2 DESCRIPTION

The backup `status` command shows if the NetBackup client has started or stopped on the Veritas Access nodes. If the NetBackup client has currently started and is running, it also shows if there are any ongoing backup or restore jobs.

4.6.3 OPTIONS

`status`

Display the status of current backup activities.

4.6.4 EXAMPLES

Display the status when no backup jobs are running.

```
Backup> status
Virtual IP state                : up
Backup service online node      : node_01
NetBackup client state          : online
NetBackup SAN client state      : online

No backup or restore jobs running.
```

Display the status when backup jobs are running that involve file systems using the NetBackup client.

```
Backup> status

Virtual IP state                : up
Backup service online node      : node_01
NetBackup client state          : online
NetBackup SAN client state      : online
```

Some file systems are busy performing backup or restore jobs by the NetBackup client

4.6.5 SEE ALSO

backup(1), netbackup(1), virtual-ip(1), virtual-name(1), show(1), start(1), stop(1)

4.7 stop

4.7.1 SYNOPSIS

`stop`

4.7.2 DESCRIPTION

The backup `stop` command stops processes that handle backup and restore. Nothing stops if there are any backup jobs running that involve file systems on the Veritas Access nodes. It will also offline the virtual IP address configured by using the backup `virtual-ip` command.

4.7.3 OPTIONS

`stop`

Stop processes that handle backup and restore. It also places the virtual IP address in the offline mode.

4.7.4 EXAMPLES

Stop processes that handle backup and restore when backup jobs are running.

```
Backup> stop
ACCESS backup ERROR V-288-0 Cannot stop, some backup jobs are running.
```

4.7.5 SEE ALSO

backup(1), netbackup(1), virtual-ip(1), virtual-name(1), show(1), status(1), start(1)

4.8 uninstall

4.8.1 SYNOPSIS

`uninstall`

4.8.2 DESCRIPTION

The backup `uninstall` command uninstalls the currently installed version of the Netbackup client.

4.8.3 EXAMPLES

Uninstall the currently installed Netbackup client.

```
Backup> uninstall
100% [#] NBU Client uninstallation completed
```

4.8.4 SEE ALSO

backup(1), netbackup(1), virtual-ip(1), virtual-name(1), show(1), status(1), stop(1), install(1), uninstall(1)

4.9 virtual-ip

4.9.1 SYNOPSIS

`virtual-ip set` *ipaddr* [*device*]

`virtual-ip reset`

4.9.2 DESCRIPTION

The backup `virtual-ip` command configures the built-in NetBackup client to use *ipaddr* as its virtual IP address and *device* as the network interface for the virtual IP.

4.9.3 OPTIONS

ipaddr Virtual IP address to be used by the NetBackup client on the Veritas Access nodes. Make sure that *ipaddr* can be resolved back to the hostname, configured by using the backup `virtual-name` command.

`virtual-ip set` *ipaddr* [*device*] Configure the NetBackup client on the Veritas Access nodes to use *ipaddr* as its highly available virtual IP address and *device* as the network interface for the virtual IP address.

A network mask can be provided with the IP address as the prefix length for the *ipaddr*. For example, `virtual-ip set 10.209.107.151/22`.

IPv6 addresses can be specified as `virtual-ip set fec0::20c:29ff:fe8e:2203/64`.

4.9.4 EXAMPLES

Configure or change the virtual IP address of the NetBackup client on the Veritas Access nodes.

```
Backup> virtual-ip set 10.10.10.10/20 pubeth1
Success.
```

Reset the virtual IP address configured for backup.

```
Backup> virtual-ip reset
Success.
```

To configure an IPv6 address for backup.

```
Backup> virtual-ip set fec0::20c:29ff:fe8e:2203/64 pubeth0
Success.
```

4.9.5 SEE ALSO

backup(1), netbackup(1), virtual-name(1), show(1), status(1), start(1), stop(1)

4.10 virtual-name

4.10.1 SYNOPSIS

`virtual-name set` *name*

`virtual-name reset`

4.10.2 DESCRIPTION

The backup `virtual-name` command configures the NetBackup client installation on the Veritas Access nodes to use *name* as its hostname.

4.10.3 OPTIONS

name Hostname to be used by a NetBackup client installation on the Veritas Access nodes. Make sure that *name* can be resolved through DNS and its IP address can be resolved back to *name* through a DNS reverse lookup. Also make sure that *name* resolves to the IP address configured through the backup `virtual-ip` command.

`virtual-name set` *name* Configure a NetBackup client installation on the Veritas Access nodes to use *name* as its hostname. Make sure that name resolves to the IP address that is configured using the `virtual-ip` command.

4.10.4 EXAMPLES

Configure or reset the hostname of the NetBackup client installation on the Veritas Access nodes.

```
Backup> virtual-name set nbuclient.veritas.com
Success.


Backup> virtual-name reset
Success.
```

4.10.5 SEE ALSO

backup(1), netbackup(1), virtual-ip(1), show(1), status(1), start(1), stop(1)

5 CIFS Commands

5.1 cifs

5.1.1 SYNOPSIS

`server` start|stop|status

`share add` *filesystem sharename* [*@virtual_ip*] [*cifsoptions*]

`share modify` *sharename* [*@virtual_ip*] [*cifsoptions*]

`share delete` *sharename* [*@virtual_ip*]

`share show` [*sharename*]

`share allow` *sharename @group1* [*,@group2,user1,user2,...*]

`share deny` *sharename @group1* [*,@group2,user1,user2,...*]

`set domaincontroller` *servernames*

`set domainuser` *username*

`set domain` *domainname*

`set workgroup` *workgroup*

`set allow_trusted_domains` *yes|no* [*trusted_domains*]

`set ntlm_auth` *yes|no*

`set security` *ads|user*

`set idmap_backend ldap` [*idmap_ou*] [*uid_range*]

`set idmap_backend rid` [*uid_range*]

`set idmap_backend hash`

`set idmap_backend ad` [*uid_range*]

`set homedirfs` [*filesystemlist*] [full_acl]

`set aio_size` *size*

`set data_migration` *yes|no*

`set clustering_mode` *normal|ctdb*

`local user add` *username* [*grouplist*]

`local password` *username*

`local user delete` *username*

`local user show` [*username*]

`local group add` *groupname*

`local group delete` *groupname*

`local group show` [*groupname*]

`local user members` *username grouplist*

`mapuser add` *CIFSusername domainname NFSusername*

`mapuser remove` *CIFSusername* [*domainname*]

`mapuser show` [*CIFSusername*] [*domainname*]

`homedir set` *username* [*domainname*] [*fs_name*]

`homedir delete` *username* [*domainname*]

`homedir deleteall`

`homedir show` [*username*] [*domainname*]

`show`

5.1.2 DESCRIPTION

The `cifs` commands maintain the current table of exported file systems for CIFS.

For the CIFS service to work properly in an active directory domain environment, the following protocals and firewall ports need be allowed or opened in the environment to allow the CIFS server to communicate smoothly with the active diretory domain controller(s) and Windows CIFS clients.

`ICMP` **:** Internet Control Message Protocol (ICMP) protocol must be allowed through the firewall from the CIFS server to the domain controllers. Enable "Allow incoming echo request" is a must requirement for running the CIFS service.

Other ports and protocols:

```
Port                Protocol
----                --------
TCP and UDP 53      DNS
TCP and UDP 88      Kerberos
TCP 139             DFSN, NetBIOS Session Service, NetLog
TCP and UDP 445     SMB,CIFS,DFSN, LSARPC, NbtSS, NetLogonR, SamR, SrvSvc
TCP and UDP 464     Kerberos change/set password
TCP 3268            LDAP GC
```

If LDAP with SSL is required, the following ports need to be opened:

```
Port                Protocol
----                --------
TCP 636             LDAP SSL
TCP 3269            LDAP GC SSL
```

5.1.3 OPTIONS

cifsoptions Comma-separated list of export options for a CIFS share are: ro, rw, guest, noguest, oplocks, nooplocks, full_acl, no_full_acl, enable_encryption, disable_encryption, shadow_copy, hide_unreadable, veto_sys_files, enable_durable_handles, owner=ownername, group=groupname, fs_mode=fspermission, dir_mask=dirpermission, create_mask=filepermission, allow=user+@group, deny=user+@group, max_connections=connections. The default export options are: ro, noguest, oplocks, no_full_acl, fs_mode=1777, dir_mask=775, create_mask=775, allow=all.

`server` start|stop|status Start, stop, or display the status of the CIFS resources. All variable settings made using the set command come into effect only when the server is restarted. For example, set can be used to set the security to ads or to a value other than ads, but the server joins or leaves the Active Directory domain only when the server is restarted after using the set command. One exception is the case when the server is already stopped, and later security is set to a value other than ads.

`share add` *filesystem sharename* [*@virtual_ip*] [*cifsoptions*] Export the file system with the given *sharename*. The new options are updated after the command is run. In `ctdb` clustering mode, you can give the directory path instead of the file system. While specifying the directory path, make sure it always starts with the file system name (Note: not with the mount point `/vx`). If *@virtual_ip* is specified, the share can only be accessed by virtual_ip.

`share modify` *sharename* [*@virtual_ip*] [*cifsoptions*] Re-export the file system with the given *sharename*. The new options are updated after the command is run. In `ctdb` clustering mode, you can give the directory path instead of the file system. While specifying the directory path, make sure it always starts with the file system name (Note: not with the mount point `/vx`). If *@virtual_ip* is specified, the share can only be accessed by virtual_ip.

`share delete` *sharename* Unexport the share with the name *sharename*.

`share show` [*sharename*] List all the exported resources. If *sharename* is specified, then Veritas Access prints all the details of the given *sharename*.

`share allow` *sharename @group1* [*,@group2,user1,user2,...*] Allow only the specified users and groups to access the share. If *all* is specified, then default access restrictions are restored on the share. User or group separator can be a comma or a plus symbol, double backslash should be added between domain name and user or group in case the allowed one comes from the trusted domain.

`share deny` *sharename @group1* [*,@group2,user1,user2,...*] Deny the specified users and groups to access the share. If *all* is specified, then all the users and groups are not able to access the share. User or group separator can be a comma or a plus symbol, double backslash should be added between domain name and user or group in case the denied one comes from the trusted domain.

`set domaincontroller` *servernames* Set the comma-separated list of domain controller server names. A domain controller can either be an Active Directory server or a Windows NT 4.0 domain controller.

`set domainuser` *username* Set the name of the domain-user. This username is used for authentication for the domain join operation.

`set domain` *domainname* Set the domain name. This also sets the WORKGROUP of the system.

`set workgroup` *workgroup* Set the workgroup name. If name of the WORKGROUP or NetBIOS domain name is different from the domain name, use this command to set the `workgroup` name.

`set allow_trusted_domains` *yes|no* [*trusted_domains*] If allow_trusted_domains is set to `yes`, then multiple domain environments are allowed. Otherwise multiple domain environments are not allowed. Note: This option cannot be set to yes if security mode is user. User can also specify the trusted domains that allow access to the CIFS server when this option is set to yes and the idmap

backend is set to `rid` or `ad`. By default, all the trusted domains are included in the configuration. Default allow_trusted_domains value: `no`

set ntlm_auth ***yes|no*** If ntlm_auth is set to no, then an NTLMv2 response needs to be sent by the client. Otherwise Veritas Access attempts to authenticate users using the NTLM-encrypted password response. Default ntlm_auth value: `yes`

set security ***ads|user*** Set CIFS security to user (user-level security that a client must log on with a valid username and password), or ads (CIFS server acts as a domain member in an Active Directory). Before setting the security to `ads`, it is required to set `domain`, `domaincontroller`, and `domainuser`. When the user executes the `server start` command, CIFS server asks for the password corresponding to the `domainuser` to join in the AD. If using security as `ads`, make sure that the clock of the AD server and the CIFS server is set to the same time. Use the NTP server if possible. Default security Value: `user`

set idmap_backend ldap [*idmap_ou*] [*uid_range*] This option tells the CIFS server to obtain SID to UID/GID mappings from a common LDAP backend. The LDAP server used for this has to be configured through the lLDAP sub-section in the network section. Trusted domains are allowed if `allow_trusted_domains` is set to `yes`. The `idmap_ou` is optional and set to `cifsidmap`. By default, the user can specify if the CIFS idmap OrganizationalUnitName(OU) is named differently on the LDAP server. The `uid_range` is optional and set to `10000-1000000` by default.

set idmap_backend rid [*uid_range*] User can use this option to get the unique SID to UID/GID mappings based on RID and LOW_RANGE_ID. Trusted domains are allowed if `allow_trusted_domains` is set to `yes`. The `uid_range` is set to `10000-1000000` by default. Change it in case there are more than 1,000,000 users existing on the local cluster, joined active directory, or trusted domains. Note: Do not attempt to modify LOW_RANGE_ID (10000) if there is user data on the CIFS server, it may lead to data access denied for the UID changes.

set idmap_backend hash User can use this option to get the unique SID to UID/GID mappings by the implemented hashing algorithm. Trusted domains are allowed if `allow_trusted_domains` is set to `yes`.

set idmap_backend ad [*uid_range*] User can use this option to get the unique UID/GID from the domain by reading id mappings from an AD server that uses RFC2307/SFU schema extensions, this is a readonly idmap backend. Trusted domains are allowed if `allow_trusted_domains` is set to `yes`. A valid user from a domain or a trusted domain should have a UID as well as a GID for the user's primary group. The `uid_range` is set to `10000-1000000` by default, change it in case there are more than 1,000,000 users existing on the local cluster, joined active directory, or trusted domains. Note: ID range is adjusted automatically according to the search results of the defined UNIX IDs from the domain after a CIFS server restart.

set homedirfs [*filesystemlist*] [full_acl] Set the given list of file systems to be used for the home directory. All of the home directories of the users are created in these file systems. *filesystem* is a comma-separated list of file system names. full_acl can be set as an option. Automatic migration of the content of users (i.e., users home directories) from one file system to another file system while switching homedir is not supported. So if an administrator changes homedir from fs1 to fs2, then home directories of the user do not migrate from fs1 to fs2 automatically.

set aio_size ***size*** Set aio_fork option. If *size* is not 0, then enable aio_fork and set it as aio read/write size. If it's 0, then disable aio_fork and set 0 to aio read/write size.

set data_migration ***yes|no*** User can use this option to enable data migration from a Windows machine by a domain administrator when ACL information for files or directories should be preserved. The ROBOCOPY command from the Windows Resource Kit Tools is required to complete the task. Set the option to `yes` and restart the CIFS server, and then data migration mode is enabled. Set the option to `no` after data migration is completed for CIFS server security.

set clustering_mode ***normal|ctdb*** Set clustering mode for the CIFS server. There are two clustering modes avilable with Veritas Access: *normal* and *ctdb*. Each clustering mode supports all of the three operating modes. The CTDB clustering mode is a different clustered implementation of Veritas Access CIFS, which supports all the features that are supported by normal clustering mode as well as some additional features. Veritas Access supports automated and seamless migration of shares and home directories from one mode to another.

local user add ***username*** **[*grouplist*]** Add the local CIFS user. This user gets authenticated when security is set to user.

local password ***username*** Reset the password for the local CIFS user.

local user delete ***username*** Delete the specified CIFS user.

local user show [*username*] Show UID and group(s) details of the given username.

local group add ***groupname*** Add the local CIFS group.

local group delete ***groupname*** Delete the local CIFS group.

local group show [*groupname*] Show the list of available local CIFS groups. If 'groupname' is specified, it shows all the users belonging to that group.

local user members ***username grouplist*** Add the local CIFS user to the given groups in the grouplist.

mapuser add ***CIFSusername domainname NFSusername*** Map CIFS user to NFS user. This mapping is useful when the same file system is accessed using both CIFS and NFS. This functionality can also be used to map multiple users to a single username to share common data.

mapuser remove ***CIFSusername*** **[*domainname*]** Remove the mapping between CIFS and NFS user. It may be possible that the CIFS user may not be able to access its previous data after removing the mapping. Default domain is `local`.

mapuser show [*CIFSusername*] [*domainname*] Show the mapping between CIFS and NFS user. If CIFS username is not specified, it shows all the existing mappings. Default domain name is `local`.

homedir set ***username*** **[*domainname*] [*fs_name*]** Set the homedir for the given user. If the home directory does not exist for the given user, this command creates that user's home directory. Use *fs_name* to specify the home directory file system where the user's home directory is created. Otherwise, the user's home directory is created on the home directory file system that has the fewest home directories. Use `storage quota cifshomedir set` to set the quota vaule for the user, otherwise the value set from the `storage quota cifshomedir setdefault` is used to configure the quota limit. If either user or default quota is not set, 0 is taken as the default value that is treated as the unlimited quota.

homedir delete ***username*** **[*domainname*]** Delete the home directory of the given user.

homedir deleteall Delete all of the home directories on the homedir file system.

homedir show [*username*] [*domainname*] Display homedir usage information.

show Display the list of all the global options and their values. Use the `set` command to modify these values.

5.1.4 EXAMPLES

Add a local CIFS group `grp1`

```
CIFS> local group add grp1
Adding GROUP : grp1
Success: Group grp1 created successfully
```

Create the new CIFS user `usr1` and assign it to existing groups `grp1`.

```
CIFS> local user add usr1 grp1
Input password for usr1.
Enter password:
Re-enter password:
Adding USER : usr1
Success: User usr1 created successfully
```

Reset password for the local CIFS user `usr1`.

```
CIFS> local password usr1
Changing password for usr1.
New password:
Re-enter new password:
Password changed for user: 'usr1'
```

Delete the local CIFS user `usr1`

```
CIFS> local user delete usr1
Deleting User: usr1
Success: User usr1 deleted successfully
```

Show the list of local CIFS users.

```
CIFS> local user show
List of Users
-------------
usr1
```

Map CIFS user to NFS user.

```
CIFS> mapuser add administrator VERITASDOMAIN.COM nfsusr1
```

Remove the mapping between CIFS and NFS user.

```
CIFS> mapuser remove administrator VERITASDOMAIN.COM
```

Show the mapping between CIFS and NFS user.

```
CIFS> mapuser show  administrator VERITASDOMAIN.COM
```

```
CIFSUserName    DomainName      NFSUserName
administrator   VERITASDOMAIN   nfsusr1
```

Export the file system `fs1` with name `share1`.

```
CIFS> share add fs1 share1
Exporting CIFS filesystem : share1 ...
```

Display the list of shares exported over CIFS.

```
CIFS> share show
       ShareName    FileSystem  ShareOptions
          share1           fs1  owner=root,group=root
```

Unexport the share with the name `share1`:

```
CIFS> share delete share1
Unexporting CIFS share : share1 ....
```

Allow user `user1` and group `group1` to access the share `share1`:

```
CIFS> share allow share1 user1,@group1
Warning: Modifying an already existing share.
.........Done
```

Allow trusted domain user `user1` and group `group1` to access the share `share1`:

```
CIFS> share allow share1 domain\\\\\\\\user1,@domain\\\\\\\\group1
Warning: Modifying an already existing share.
.........Done
```

Deny user `user1` and group `group1` to access the share `share1`:

```
CIFS> share deny share1 user1,@group1
Warning: Modifying an already existing share.
.........Done
```

Deny trusted domain user `user1` and group `group1` to access the share `share1`:

```
CIFS> share deny share1 domain\\\\\\\\user1,@domain\\\\\\\\group1
Warning: Modifying an already existing share.
.........Done
```

Setting the domain parameters for joining an active directory or NT domain. In this example, the name of the domain is set to `VERITASDOMAIN.COM`, domain user is set to `administrator`, and domain controller for the domain `VERITASDOMAIN.COM` is set to `VERITASSERVER`:

```
CIFS> set domain ``VERITASDOMAIN.COM``
Global option updated. Note: Restart the CIFS server.
CIFS> set domainuser ``administrator``
Global option updated. Note: Restart the CIFS server.
CIFS> set domaincontroller ``VERITASSERVER``
Global option updated. Note: Restart the CIFS server.
```

After setting all of the above domain parameters, the user can set the security to either `ads` if the domain controller is an active directory server, or the user can set the security to `domain` if the domain controller is an NT domain controller:

```
CIFS> set security ``ads``
Global option updated. Note: Restart the CIFS server.
```

Display the list of global options:

```
    CIFS> show
                          Name  Value
                          ----  -----
                   netbios name  test
```

```
                     ntlm auth  yes
         allow trusted domains  no
                     homedirfs
                      aio size  0
                 idmap backend  rid:10000-1000000
                     workgroup  VERITASDOMAIN
                      security  ads
                        Domain  VERITASDOMAIN.COM
                   Domain user  administrator
             Domain Controller  VERITASSERVER
               Clustering Mode  normal

Note that the system will make a note of these parameters, but it will not attempt to␣
↪join the specified domain. That will happen only when the CIFS server is started␣
↪the next time using the ``server`` command.

Starting the CIFS server after all domain-related parameters have been set.
    CIFS> server start

    The skew of the system clock with respect to Domain controller is:  3 seconds

    Time on Domain controller : Fri November 30 06:00:03 2008
    Time on this system : Fri November 30 06:00:00 PDT 2008

    If the above clock skew is greater than that allowed by the server,
    then the system won't be able to join the AD domain

    Enter password for user 'administrator':

    Trying to become a member in AD domain VERITASDOMAIN.COM ...
    Joined domain VERITASDOMAIN.COM OK
    Starting CIFS Server..

Display the status of the CIFS server.

CIFS> server status
CIFS Status on test_01 : ONLINE
CIFS Status on test_02 : ONLINE

Homedirfs                 : fs
Security                  : ads
Domain membership status  : Enabled
Domain                    : VERITASDOMAIN.COM
Domain Controller         : VERITASSERVER
Domain User               : administrator
Clustering Mode           : normal
```

Display the status of the CIFS server if allowed_trusted_domain is set to yes.

```
CIFS> server status
CIFS Status on test_01 : ONLINE
CIFS Status on test_02 : ONLINE

Homedirfs                 : fs
Security                  : ads
Domain membership status  : Enabled
Domain                    : VERITASDOMAIN.COM
Domain Controller         : VERITASSERVER
```

```
Domain User                : administrator
Clustering Mode            : normal
Trusted Domains            : VERITASDOMAIN1 [VERITASDOMAIN2] VERITASDOMAIN3
```

Note: Domain name with square bracket means this trusted domain is obsolete.

Stop the CIFS server.

```
CIFS> server stop
Stopping CIFS Server.....Success.
```

Exporting a directory `fs1/dir1` in `ctdb` clustering mode.

```
ctdb.CIFS> share add fs1/dir1 share1 rw,full_acl
Exporting CIFS filesystem : share1 ..Success.

ctdb.CIFS> share show
      ShareName    FileSystem     ShareOptions
         share1    fs1/dir1       owner=root,group=root,fs_mode=755,rw,full_acl
```

Migration from `ctdb` clustering mode to `normal` clustering mode.

1. Check the status of the CIFS server.

```
ctdb.CIFS> server status
CIFS Status on ctdb_01 : ONLINE
CIFS Status on ctdb_02 : ONLINE

Security                   : ads
Domain membership status   : Enabled
Domain                     : VERITASDOMAIN.COM
Workgroup                  : VERITASDOMAIN
Domain Controller          : veritasdomain.veritas.com
Domain User                : administrator
Clustering Mode            : ctdb
```

2. List the shares and homedir.

```
ctdb.CIFS> share show
      ShareName                FileSystem  ShareOptions
         share1    fs1/Veritas Access  owner=root,group=root,fs_mode=755,rw,noguest
         share2    fs1/Veritas Access  owner=root,group=root,fs_mode=755,ro,guest
         share3                       fs3  owner=root,group=root,fs_mode=1777
         share4                       fs4  owner=root,group=root,fs_mode=1777,rw

ctdb.CIFS> homedir show
                 UserName      DomainName
                     usr1           LOCAL
            administrator           VERITASDOMAIN
```

3. Stop the CIFS server and set clustering mode to `normal`.

```
ctdb.CIFS> server stop

Disabling membership in AD domain VERITASDOMAIN.COM

Enter password for user 'administrator':
Left AD domain VERITASDOMAIN.COM
```

```
Stopping CIFS Server......Success.

ctdb.CIFS> set clustering_mode normal
Global option updated. Note: Restart the CIFS server.
```

4. Start the CIFS server in `normal` clustering mode.

```
ctdb.CIFS> server start
Uninstalling 'ctdb' Clustering Mode..............Success.
Installing 'normal' Clustering Mode..........Success.

The skew of the system clock with respect to Domain controller veritasdomain.veritas.
↪com
(10.209.110.210) is:  9 seconds

Time on Domain controller : Thu Aug 19 16:54:03 2010
Time on this system : Thu Aug 19 16:53:54 IST 2010

If the above clock skew is greater than that allowed by the server,
then the system won't be able to join the AD domain


Trying to become a member in AD domain VERITASDOMAIN.COM ...

Enter password for user 'administrator':
Joined domain VERITASDOMAIN.COM OK
ACCESS cifs WARNING V-288-0 Migration of following shares are not supported in normal
clustering mode
              Sharename   FS Name
                 share1   fs1/Veritas Access
                 share2   fs1/Veritas Access
Starting CIFS Server.....Success.
```

5. Check the shares and home directory after migrating to `normal` clustering modes.

```
ctdb.CIFS> share show
      ShareName             FileSystem  ShareOptions
         share3                    fs3  owner=root,group=root,fs_mode=1777
         share4                    fs4  owner=root,group=root,fs_mode=1777,rw

ctdb.CIFS> homedir show
                 UserName      DomainName
                     usr1           LOCAL
            administrator           VERITASDOMAIN
```

`Note:` You can see that directory level share cannot be migrated to normal clustering mode. Rest of the share and home directory are migrated perfectly.

Migration from `normal` clustering mode to `ctdb` clustering mode.

1. Check the server status to confirm that the current clustering mode is `normal`.

```
ctdb.CIFS> server status
CIFS Status on ctdb_01 : ONLINE
CIFS Status on ctdb_02 : ONLINE

Security                  : ads
Domain membership status  : Enabled
Domain                    : VERITASDOMAIN.COM
```

```
Workgroup                  : VERITASDOMAIN
Domain Controller          : veritasdomain.veritas.com
Domain User                : administrator
Clustering Mode            : normal
```

2. List the shares and the homedir.

```
ctdb.CIFS> share show
      ShareName             FileSystem  ShareOptions
         share3                    fs3  owner=root,group=root,fs_mode=1777
         share4                    fs4  owner=root,group=root,fs_mode=1777,rw

ctdb.CIFS> homedir show
                 UserName      DomainName
                     usr1           LOCAL
            administrator           VERITASDOMAIN
```

3. Now stop the CIFS server and change the clustering mode to the `ctdb` mode.

```
ctdb.CIFS> server stop
Stopping CIFS Server.....Success.

ctdb.CIFS> set clustering_mode ctdb
Global option updated. Note: Restart the CIFS server.
```

4. Start the CIFS server in `ctdb` clustering mode and check the server status.

```
ctdb.CIFS> server start

Disabling membership in AD domain VERITASDOMAIN.COM

Enter password for user 'administrator':
Left AD domain VERITASDOMAIN.COM
Uninstalling 'normal' Clustering Mode..........Success.
Installing 'ctdb' Clustering Mode.........Success.
Starting CIFS Server....
The skew of the system clock with respect to Domain controller veritasdomain.veritas.
↪com
(10.209.110.210) is: 9 seconds

Time on Domain controller : Thu Aug 19 17:07:19 2010
Time on this system : Thu Aug 19 17:07:10 IST 2010

If the above clock skew is greater than that allowed by the server,
then the system won't be able to join the AD domain


Trying to become a member in AD domain VERITASDOMAIN.COM ...

Enter password for user 'administrator':
Joined domain VERITASDOMAIN.COM OK
..Success.
ctdb.CIFS> server status
CIFS Status on ctdb_01 : ONLINE
CIFS Status on ctdb_02 : ONLINE

Security                   : ads
Domain membership status   : Enabled
```

```
Domain                    : VERITASDOMAIN.COM
Workgroup                 : VERITASDOMAIN
Domain Controller         : veritasdomain.veritas.com
Domain User               : administrator
Clustering Mode           : ctdb
```

5. Verify that all shares and home directory are properly migrated to the `ctdb` clustering mode.

```
ctdb.CIFS> share show
      ShareName    FileSystem  ShareOptions
         share1           fs1  owner=root,group=root,fs_mode=1777,rw,full_acl
         share3           fs3  owner=root,group=root,fs_mode=1777
         share4           fs4  owner=root,group=root,fs_mode=1777,rw

ctdb.CIFS> homedir show
                 UserName      DomainName
                     usr1           LOCAL
            administrator           VERITASDOMAIN
```

5.1.5 SEE ALSO

server(1), share(1)

5.2 homedir

5.2.1 SYNOPSIS

`homedir set` *username* [*domainname*] [*fs_name*]

`homedir delete` *username* [*domainname*]

`homedir deleteall`

`homedir show` [*username*] [*domainname*]

5.2.2 DESCRIPTION

The `homedir` commands manage the home directories for the users.

5.2.3 OPTIONS

`homedir set` *username* [*domainname*] [*fs_name*] Set the homedir for the given user. If the home directory does not exist for the given user, this command creates that user's home directory. Use *fs_name* to specify the home directory file system where the user's home directory is created. Otherwise, the user's home directory is created on the home directory file system that has the fewest home directories. Use `storage quota cifshomedir set` to set the quota vaule for the user, otherwise the value set from the `storage quota cifshomedir setdefault` is used to configure the quota limit. If either user or default quota is not set, 0 is taken as the default value, which is treated as an unlimited quota.

`homedir delete` *username* [*domainname*] Delete the home directory of the given user.

`homedir deleteall` Delete all of the home directories on the homedir file system.

`homedir show` [*username*] [*domainname*] Display the homedir usage information.

5.2.4 EXAMPLES

Set the homedir for the given user.

```
CIFS> homedir set usr1 local
CIFS>
```

Delete the home directory of user2.

```
CIFS> homedir delete user2
Do you want to delete homedir for user user2 (y/n): y
Success: Home directory deleted for CIFS local user: user2
```

Display the home directory usage information.

```
CIFS> homedir show
UserName      DomainName
user1         LOCAL
user2         LOCAL
```

5.2.5 SEE ALSO

set(1)

5.3 local

5.3.1 SYNOPSIS

`local user add` *username* [*grouplist*]

`local password` *username*

`local user delete` *username*

`local user show` [*username*]

`local group add` *groupname*

`local group delete` *groupname*

`local group show` [*groupname*]

`local user members` *username grouplist*

5.3.2 DESCRIPTION

The `local` commands add, delete, or display CIFS user(s)/group(s).

5.3.3 OPTIONS

grouplist Comma-separated list of group names.

`local user add` ***username*** **[*grouplist*]** Add the local CIFS user. This user gets authenticated when security is set to user.

`local password` ***username*** Reset password for the local CIFS user.

`local user delete` ***username*** Delete the specified CIFS user.

`local user show` **[*username*]** Show UID and group(s) details of the given username.

`local group add` ***groupname*** Add the local CIFS group.

`local group delete` ***groupname*** Delete the local CIFS group.

`local group show` **[*groupname*]** Show the list of available local CIFS groups. If 'groupname' is specified, it shows all the users belonging to that group.

`local user members` ***username grouplist*** Add the local CIFS user to the given groups in the grouplist.

5.3.4 EXAMPLES

Add a local CIFS group `grp1`.

```
CIFS> local group add grp1
Adding GROUP : grp1
Success: Group grp1 created successfully
```

Create the new CIFS user `usr1` and assign it to existing groups `grp1`.

```
CIFS> local user add usr1 grp1
Input password for usr1.
Enter password:
Re-enter password:
Adding USER : usr1
Success: User usr1 created successfully
```

Reset password for the local CIFS user `usr1`.

```
CIFS> local password usr1
Changing password for usr1.
New password:
Re-enter new password:
Password changed for user: 'usr1'
```

Show the list of local CIFS users.

```
CIFS> local user show
List of Users
-------------
usr1
```

Delete the local CIFS user `usr1`.

```
CIFS> local user delete usr1
Deleting User: usr1
Success: User usr1 deleted successfully
```

5.3.5 SEE ALSO

server(1)

5.4 mapuser

5.4.1 SYNOPSIS

`mapuser add` *CIFSusername domainname NFSusername*

`mapuser remove` *CIFSusername* [*domainname*]

`mapuser show` [*CIFSusername*] [*domainname*]

5.4.2 DESCRIPTION

The `mapuser` commands add, remove, or display the mapping between CIFS and NFS users. Typically a *CIFSusername* is a user coming from the AD server (*domainname*), or a locally created CIFS user on this system (*local*). A *NFSusername* is a user coming from a locally created CIFS user on this system or from a NIS/LDAP server configured in the network section. To make sure user mappings work properly with a NIS/LDAP server, the nsswitch setting may need to be adjusted in the network section.

5.4.3 OPTIONS

`mapuser add` *CIFSusername domainname NFSusername* Map CIFS user to NFS user. This mapping is useful when the same file system is accessed using both CIFS and NFS. This functionality can also be used to map multiple users to a single username to share common data. Note: If *** is specified both for the CIFS user and the NFS user, mappings are created automatically between the same name user between CIFS and NFS users on the CIFS server when such user logs on.

`mapuser remove` *CIFSusername* [*domainname*] Remove the mapping between CIFS and NFS user. It may be possible that the CIFS user may not be able to access its previous data after removing the mapping. Default domain is `local`. Note: If *** is specified, no more mappings are created between the same name CIFS and NFS users, however the already created same name mappings are left in the user mapping list on the CIFS server.

`mapuser show` [*CIFSusername*] [*domainname*] Show the mapping between the CIFS and the NFS user. If the CIFS username is not specified, it shows all the existing mappings. Default domain name is `local`.

5.4.4 EXAMPLES

Map CIFS user to NFS user.

```
CIFS> mapuser add administrator VERITASDOMAIN.COM nfsusr1
```

Add mappings between all the same name CIFS and NFS users.

```
CIFS> mapuser add * VERITASDOMAIN.COM *
```

Remove the mapping between the CIFS and the NFS user.

```
CIFS> mapuser remove administrator VERITASDOMAIN.COM
```

Remove the mappings between the same name CIFS and NFS users.

```
CIFS> mapuser remove * VERITASDOMAIN.COM
```

Show the mappings between the CIFS and the NFS user.

```
CIFS> mapuser show  administrator VERITASDOMAIN.COM

CIFSUserName    DomainName      NFSUserName
administrator   VERITASDOMAIN   nfsusr1
```

5.4.5 SEE ALSO

share(1)

5.5 server

5.5.1 SYNOPSIS

`server` [start|stop|status]

5.5.2 DESCRIPTION

Start, stop, or display the status of the CIFS resources. All variable settings made using the set command come into effect only when the server is restarted. For example, set can be used to set the security to ads or to a value other than ads, but the server joins or leaves the active directory domain only when the server is restarted after issuing the set command. One exception is the case when the server is already stopped, and later security is set to a value other than ads.

5.5.3 OPTIONS

`server` [start|stop|status] Start, stop, or check the status of the CIFS resources.

5.5.4 EXAMPLES

Starting the CIFS server after all domain-related parameters have been set.

```
CIFS> server start

The skew of the system clock with respect to Domain controller is:  3 seconds

Time on Domain controller : Fri November 30 06:00:03 2008
Time on this system : Fri November 30 06:00:00 PDT 2008

If the above clock skew is greater than that allowed by the server,
then the system won't be able to join the AD domain

Enter password for user 'administrator':

Trying to become a member in AD domain VERITASDOMAIN.COM ...
Joined domain VERITASDOMAIN.COM OK
Starting CIFS Server..
```

Display the status of CIFS server.

```
CIFS> server status
CIFS Status on test_01 : ONLINE
CIFS Status on test_02 : ONLINE

Homedirfs                 : fs
Security                  : ads
Domain membership status  : Enabled
Domain                    : VERITASDOMAIN.COM
Domain Controller         : VERITASSERVER
Domain User               : administrator
Clustering Mode           : normal
```

Display the status of the CIFS server if `allowed_trusted_domain` is set to yes.

```
CIFS> server status
CIFS Status on test_01 : ONLINE
CIFS Status on test_02 : ONLINE

Homedirfs                 : fs
Security                  : ads
Domain membership status  : Enabled
Domain                    : VERITASDOMAIN.COM
Domain Controller         : VERITASSERVER
Domain User               : administrator
Clustering Mode           : normal
Trusted Domains           : VERITASDOMAIN1 [VERITASDOMAIN2] VERITASDOMAIN3
```

Note: Domain name with square brackets means this trusted domain is obsolete.

Stop the CIFS server.

```
CIFS> server stop
Stopping CIFS Server.....Success.
```

5.5.5 SEE ALSO

set(1), share(1)

5.6 set

5.6.1 SYNOPSIS

`set domaincontroller` *servernames*

`set domainuser` *username*

`set domain` *domainname*

`set workgroup` *workgroup*

`set allow_trusted_domains` *yes|no* [*trusted_domains*]

`set ntlm_auth` *yes|no*

`set security` *ads|user*

`set idmap_backend ldap` [*idmap_ou*] [*uid_range*]

`set idmap_backend rid` [*uid_range*]

`set idmap_backend hash`

`set idmap_backend ad` [*uid_range*]

`set homedirfs` [*filesystemlist*] [full_acl]

`set aio_size` *size*

`set data_migration` *yes|no*

`set clustering_mode` *normal|ctdb*

5.6.2 DESCRIPTION

The `set` commands allow the user to set various parameters required for CIFS functioning.

5.6.3 OPTIONS

`set domaincontroller` ***servernames*** Set the comma-separated list of domain controller server names. A domain controller can either be an active directory server or a Windows NT 4.0 domain controller.

`set domainuser` ***username*** Set the name of the domain user. This username is used for authentication for the domain join operation.

`set domain` ***domainname*** Set the domain name. This also sets the WORKGROUP of the system.

`set workgroup` ***workgroup*** Set the workgroup name. If name of the WORKGROUP or NetBIOS domain name is different from the domain name, use this command to set `workgroup` name.

`set allow_trusted_domains` ***yes|no*** **[*trusted_domains*]** If allow_trusted_domains is set to `yes`, then multiple domain environments are allowed. If set to `no`, multiple domain environments are not allowed. **Note:** This option cannot be set to yes if security mode is user.

The user can also specify the trusted domains that allow access to the CIFS server when this option is set to yes and the `idmap backend` is set to `rid` or `ad`. By default all the trusted domains are included in the configuration.

Default allow_trusted_domains Value: `no`

set ntlm_auth ***yes|no*** If ntlm_auth is set to `no`, then an NTLMv2 response needs to be sent by the client. If set to `yes`, it attempts to authenticate users using the NTLM-encrypted password response.

Default ntlm_auth Value: `yes`

set security ***ads|user*** Set CIFS security to user (user-level security that a client must log on with a valid username and password) or ads (CIFS server acts as a domain member in an active directory). Before setting the security to `ads`, it is required to set `domain`, `domaincontroller`, and `domainuser`. When the user executes the `server start` command, the CIFS server asks for the password corresponding to the `domainuser` to join in the AD. If using security as `ads`, make sure that the clock of the AD server and the CIFS server is set to the same time. Use the NTP server if possible. If using security as `ads`, make sure that the clock of the AD server and Veritas Access is set to the same time. Use the NTP server if possible.

Default security Value: `user`

set idmap_backend ldap [*idmap_ou*] [*uid_range*] This option tells the CIFS server to obtain SID to UID/GID mappings from a common LDAP backend. The LDAP server used for this has to be configured through the LDAP sub-section in the network section. Trusted domains are allowed if `allow_trusted_domains` is set to `yes`. The `idmap_ou` is optional and set to `cifsidmap`. By default, the user can specify it if the CIFS idmap OrganizationalUnitName(OU) is named differently on the LDAP server. The `uid_range` is optional and set to `10000-1000000` by default.

set idmap_backend rid [*uid_range*] The user can use this option to get the unique SID to UID/GID mappings based on RID and LOW_RANGE_ID. Trusted domains are allowed if `allow_trusted_domains` is set to `yes`.

The `uid_range` is set to `10000-1000000` by default, change it in cases where there are more than 1,000,000 users existing on the local Veritas Access cluster, joined active directory, or trusted domains. **Note:** Do not attempt to modify LOW_RANGE_ID (10000) if there is user data on the CIFS server, it may lead to data access denied for the UID changes.

set idmap_backend hash User can use this option to get the unique SID to UID/GID mappings by the implemented hashing algorithm. Trusted domains are allowed if `allow_trusted_domains` is set to `yes`.

set idmap_backend ad [*uid_range*] User can use this option to get the unique UID/GID from domain by reading id mappings from an AD server that uses RFC2307/SFU schema extensions, this is a readonly idmap backend. Trusted domains are allowed if `allow_trusted_domains` is set to `yes`. A valid user from a domain or a trusted domain should have a UID as well as a GID for the user's primary group.

The `uid_range` is set to `10000-1000000` by default, change it in cases where there are more than 1,000,000 users existing on the local Veritas Access cluster, joined active directory, or trusted domains. **Note:** ID range is adjusted automatically according to the search results of the defined UNIX IDs from the domain after a CIFS server restart.

set homedirfs [*filesystemlist*] [full_acl] Set the given list of file systems to be used for the home directory. All of the home directories of the users are created in these file systems. *filesystem* is a comma-separated list of file system names. full_acl can be set as an option. Automatic migration of the content of users (that is, users' home directories) from one file system to another file system while switching homedir is not supported. So if an administrator changes homedir from fs1 to fs2, then the home directories of the user do not migrate from fs1 to fs2 automatically.

set aio_size ***size*** Set aio_fork option. If *size* is not 0, then enable aio_fork and set it as aio read/write size. If it is 0, then disable aio_fork and set 0 to aio read/write size.

set data_migration ***yes|no*** The user can use this option to enable data migration from a Windows machine by a domain administrator, when ACL information for files or directories should be preserved. The ROBOCOPY command from the Windows Resource Kit Tools is required to complete

this task. Set the option to `yes` and restart the CIFS server, and then the data migration mode is enabled. Set the option to `no` after data migration completes for CIFS server security.

`set clustering_mode` ***normal|ctdb*** Set the clustering mode for the CIFS server. There are two clustering modes available with Veritas Access, *normal* and *ctdb*. Each clustering mode supports all of the three operating modes. The CTDB clustering mode is a different clustered implementation of Veritas Access CIFS, which supports all the features that are supported by normal clustering mode as well as some additional features. Additional features supported in CTDB clustering mode: `Directory-level share support`, `Multi-instance share export of a file system/directory`, and `better load balancing`. Veritas Access supports automated and seamless migration of shares and homedirectories from one mode to another. However while migrating from `CTDB` to `normal` clustering mode, some of the shares get discarded that are not supported in `normal` clustering mode.

5.6.4 EXAMPLES

Setting the domain parameters for joining an active directory or NT domain. In this example, the name of the domain is set to `VERITASDOMAIN.COM`, domain user is set to `administrator`, and domain controller for the domain `VERITASDOMAIN.COM` is set to `VERITASSERVER`.

```
CIFS> set domain ``VERITASDOMAIN.COM``
Global option updated. Note: Restart the CIFS server.
CIFS> set domainuser ``administrator``
Global option updated. Note: Restart the CIFS server.
CIFS> set domaincontroller ``VERITASSERVER``
Global option updated. Note: Restart the CIFS server.
```

After setting all the above domain parameters, you can set the security to either `ads` if the domain controller is an active directory server, or you can set the security to `domain` if the domain controller is an NT domain controller.

```
CIFS> set security ``ads``
Global option updated. Note: Restart the CIFS server.
```

Note that the system will make a note of these parameters, but it will not attempt to join the specified domain. That happens only when the CIFS server is started the next time using the `server` command.

5.6.5 SEE ALSO

server(1), share(1)

5.7 share

5.7.1 SYNOPSIS

`share add` *filesystem sharename* [*@virtual_ip*] [*cifsoptions*]

`share modify` *sharename* [*@virtual_ip*] [*cifsoptions*]

`share delete` *sharename* [*@virtual_ip*]

`share show` [*sharename*]

`share allow` *sharename @group1* [*,@group2,user1,user2,...*]

`share deny` *sharename @group1* [*,@group2,user1,user2,...*]

5.7.2 DESCRIPTION

The `share` commands add, delete, or display CIFS shares.

5.7.3 OPTIONS

cifsoptions Comma-separated list of export options for a CIFS share: {ro, rw, guest, noguest, oplocks, nooplocks, full_acl, no_full_acl, enable_encryption, disable_encryption, shadow_copy, hide_unreadable, veto_sys_files, enable_durable_handles, owner=ownername, group=groupname, fs_mode=fspermission, dir_mask=dirpermission, create_mask=filepermission, allow=user+@group, deny=user+@group,max_connections=connections}. The default export options are{ro, noguest, oplocks, no_full_acl, fs_mode=1777, dir_mask=775, create_mask=775, allow=all}.

`share add` ***filesystem sharename* [*@virtual_ip*] [*cifsoptions*]** Export the file system with the given *sharename*. The new options are updated after the command is run. In both clustering modes, you can give the directory path instead of the file system. While specifying the directory path, make sure it always starts with the file system name (Note: not with the mount point `/vx`). If *@virtual_ip* is specified, the share can only be accessed by virtual_ip.

`share modify` ***sharename* [*@virtual_ip*] [*cifsoptions*]** Re-export the file system with the given *sharename*. The new options are updated after the command is run. In both clustering modes, you can give the directory path instead of the file system. While specifying the directory path, make sure it always starts with file system name (Note: not with the mount point `/vx`). If *@virtual_ip* is specified, the share can only be accessed by virtual_ip.

`share delete` ***sharename*** Unexport the share with the name *sharename*.

`share show` **[*sharename*]** List all the exported resources. If *sharename* is specified, then it prints all the details of the given *sharename*.

`share allow` ***sharename @group1* [*,@group2,user1,user2,...*]** Allow only the specified users and groups to access the share. If *all* is specified, then default access restrictions are restored on the share. User or group separator can be a comma or a plus symbol, double backslash should be added between the domain name and the user or group in cases where the allowed one comes from the trusted domain.

`share deny` ***sharename @group1* [*,@group2,user1,user2,...*]** Deny the specified users and groups to access the share. If *all* is specified, then all the users and groups are not able to access the share. User or group separator can be a comma or a plus symbol, double backslash should be added between the domain name and the user or group in cases where the denied one comes from the trusted domain.

5.7.4 EXAMPLES

Export the file system `fs1` with name `share1`.

```
CIFS> share add fs1 share1
Exporting CIFS filesystem : share1 ...
```

Display the list of shares exported over CIFS.

```
CIFS> share show
ShareName    FileSystem  ShareOptions
share1       fs1         owner=root,group=root
```

Unexport the share with the name `share1`.

```
CIFS> share delete share1
Unexporting CIFS share : share1 ....
```

Allow user `user1` and group `group1` to access the share `share1`.

```
CIFS> share allow share1 user1,@group1
Warning: Modifying an already existing share.
.........Done
```

Allow trusted domain user `user1` and group `group1` to access the share `share1`.

```
CIFS> share allow share1 domain\\\\\\\\user1,@domain\\\\\\\\group1
Warning: Modifying an already existing share.
.........Done
```

Deny user `user1` and group `group1` to access the share `share1`.

```
CIFS> share deny share1 user1,@group1
Warning: Modifying an already existing share.
.........Done
```

Deny trusted domain user `user1` and group `group1` to access the share `share1`.

```
CIFS> share deny share1 domain\\\\\\\\user1,@domain\\\\\\\\group1
Warning: Modifying an already existing share.
.........Done
```

Exporting a directory as a CIFS share, you should switch to the `ctdb` clustering mode.

1. Check the status of the CIFS server to confirm that the current clustering mode is `ctdb`.

```
ctdb.CIFS> server status
CIFS Status on ctdb_01 : ONLINE
CIFS Status on ctdb_02 : ONLINE

Security                  : ads
Domain membership status  : Enabled
Domain                    : VERITASDOMAIN.COM
Workgroup                 : VERITASDOMAIN
Domain Controller         : veritasdomain.veritas.com
Domain User               : administrator
Clustering Mode           : ctdb
```

2. Export a directory as a CIFS share.

```
ctdb.CIFS> share add fs1/dir1 share1 rw,full_acl
Exporting CIFS filesystem : share1 ..Success.
```

3. List the shares.

```
ctdb.CIFS> share show
ShareName     FileSystem  ShareOptions
share1        fs1/dir1    owner=root,group=root,fs_mode=755,rw,full_acl
```

4. If a directory name contains a space, then it should be entered using double quotes (”). For example:

```
ctdb.CIFS> share add "fs1/dir2 a" share2 rw
Exporting CIFS filesystem : share2 ..Success.

ctdb.CIFS> share show
ShareName     FileSystem  ShareOptions
share2        fs1/dir2 a  owner=root,group=root,fs_mode=755,rw
share1        fs1/dir1    owner=root,group=root,fs_mode=755,rw,full_acl
```

5. Modifying an existing share.

```
ctdb.CIFS> share modify share2 ro,full_acl
Warning: Modifying an already existing share.
...Done

ctdb.CIFS> share show
ShareName     FileSystem  ShareOptions
share2        fs1/dir2 a  owner=root,group=root,fs_mode=1777,ro,full_acl
share1        fs1/dir1    owner=root,group=root,fs_mode=755,rw,full_acl
```

Exporting a directory to provide read access to all; but write access to the limited set of users that need to be authenticated.

```
ctdb.CIFS> share add "fs1/dir2 b" share1 rw,noguest
Exporting CIFS filesystem : share1 ..Success.

ctdb.CIFS> share add "fs1/dir2 b" share2 ro,guest
Exporting CIFS filesystem : share2 ..Success.

ctdb.CIFS> share show
ShareName     FileSystem  ShareOptions
share1        fs1/dir2 b  owner=root,group=root,fs_mode=755,rw,noguest
share2        fs1/dir2 b  owner=root,group=root,fs_mode=755,ro,guest
```

5.7.5 SEE ALSO

server(1)

5.8 show

5.8.1 SYNOPSIS

`show`

5.8.2 DESCRIPTION

The `show` command displays the list of global options and their values. These are the settings that have been configured primarily with the `set` command.

5.8.3 EXAMPLES

Display the list of global options.

```
CIFS> show

                 Name  Value
                 ----  -----
          netbios name  test
             ntlm auth  yes
 allow trusted domains  no
             homedirfs
              aio size  0
         idmap backend  rid:10000-1000000
             workgroup  VERITASDOMAIN
              security  ads
                Domain  VERITASDOMAIN.COM
           Domain user  administrator
     Domain Controller  VERITASSERVER
       Clustering Mode  normal
```

5.8.4 SEE ALSO

set(1)

6 Cluster Commands

6.1 cluster

6.1.1 SYNOPSIS

`reboot` *nodename*

`show` [`currentload`]

`shutdown` *nodename*

`add` *ipaddr1* [, *ipaddr2*,...]

`del` *nodename1* [, *nodename2*,...]

6.1.2 DESCRIPTION

The cluster commands allow you to view the nodes in the cluster and their states. You can add, remove, reboot, or shutdown nodes in the cluster configuration.

6.1.3 OPTIONS

nodename The node on which the operation takes place. A value of `all` indicates the operation takes place on all of the nodes in the cluster. The value `all` does not apply to the `del` command.

ipaddr The accessible IP address of the node to be added to the cluster.

`reboot` *nodename* Reboot a node or all of the nodes in the cluster. To reboot a node, specify the *nodename* as it is displayed in the `cluster show` command. To reboot all the nodes in the cluster use `all` for *nodename*.

`show` Display the nodes in the cluster, their states, CPU load, and network load, during the past 15 minutes. It also displays information about nodes that are being added to the cluster, deleted from the cluster and nodes on which upgrade is in progress.

`show` [*currentload*] If you specify currentload, the load statistics show the CPU load, and network load for the 5 seconds after the command is issued.

shutdown ***nodename*** Shut down a node or all of the nodes in the cluster. To shut down a node, specify the *nodename* as it is displayed in the `cluster show` command. To shut down all the nodes in the cluster, use `all` for *nodename*.

add ***ipaddr1*** **[,** ***ipaddr2*****,...]** Add one or more nodes into the cluster. The *ipaddr* should be known before `add` operation. The cluster must be running and the new nodes must have the required OS installed.

del ***nodename1*** **[,** ***nodename2*****,...]** Delete one or more nodes from the cluster. The specified nodes must be in the cluster but can not be the console node. You cannot delete all of the running nodes in the cluster. After a node is successfully deleted, the node is still accessible at the IP address until the node is rebooted. After the reboot, the node frees the used IP resources and is assigned to the original ip.

6.1.4 EXAMPLES

Display the current state of all the nodes in the cluster and their loads, during the past 15 minutes.

```
Cluster> show
Node            State      CPU(15 min)    pubeth0(15 min)    pubeth1(15 min)
                                %        rx(MB/s) tx(MB/s)  rx(MB/s) tx(MB/s)
----            -----      -----------   -------- --------  -------- --------
test_1          RUNNING          1.35       0.00     0.00      0.00     0.00
test_2          RUNNING          1.96       0.00     0.00      0.00     0.00
test_3          FAULTED

Nodes in Transition
Node/IP                    Operation           State      Description
-------                    ---------           -----      -----------
10.200.58.202              Add node            FAILED     Installing packages
test_6                     Delete node         ONGOING    Removing node
test_4,test_5              Rolling upgrade     ONGOING    Rolling upgrade phase 2
```

Display the CPU and network loads collected for the next 5 seconds.

```
Cluster> show currentload
Node            State      CPU(5 sec)     pubeth0(5 sec)     pubeth1(5 sec)
                                %        rx(MB/s) tx(MB/s)  rx(MB/s) tx(MB/s)
----            -----      -----------   -------- --------  -------- --------
test_1          RUNNING          1.35       0.00     0.00      0.00     0.00
test_2          RUNNING          1.96       0.00     0.00      0.00     0.00
test_3          FAULTED
```

Shut down a node in the cluster.

```
Cluster> shutdown test_4
Stopping Cluster processes on test_4
Sent shutdown command to test_4. SSH sessions to test_4 may terminate.
```

Shut down all nodes in the cluster.

```
Cluster> shutdown all
Stopping Cluster processes on all
SSH sessions to all nodes may terminate.
Sent shutdown command to test_2
Sent shutdown command to test_3
Sent shutdown command to test_4
Sent shutdown command to test_1
```

Reboot a node in the cluster.

```
Cluster> reboot test_4
Stopping Cluster processes on test_4
Sent reboot command to test_4. SSH sessions to test_4 may terminate.
```

Reboot all nodes in the cluster.

```
Cluster> reboot all
Stopping Cluster processes on all
SSH sessions to all nodes may terminate.
Sent reboot command to test_2
Sent reboot command to test_3
Sent reboot command to test_4
Sent reboot command to test_1
```

6.1.5 SEE ALSO

add(1), del(1), reboot(1), show(1), shutdown(1)

6.2 add

6.2.1 SYNOPSIS

`add` *ipaddr1* [, *ipaddr2*,...]

6.2.2 DESCRIPTION

The cluster `add` command adds one or more nodes into the running cluster. Provide the IP address of each new node. To specify multiple nodes, separate the IP addresses with commas. The cluster needs to be running and the new nodes must have the required OS installed.

6.2.3 EXAMPLES

Add a node 10.10.10.10 into the cluster.

```
Cluster> add 10.10.10.10
```

Add 2 nodes 10.10.10.10,10.10.10.11 into the cluster.

```
Cluster> add 10.10.10.10,10.10.10.11
```

6.2.4 SEE ALSO

del(1), reboot(1), show(1), shutdown(1)

6.3 del

6.3.1 SYNOPSIS

`del` *nodename1* [, *nodename2*,...]

6.3.2 DESCRIPTION

The cluster `del` command deletes one or more nodes from current cluster. The specified nodes must be in the cluster but cannot be the console node. You cannot delete all the running nodes in the cluster.

After a node is successfully deleted, the node is still accessible at the IP address until the node is rebooted. After the reboot, the node frees the used IP resources and is assigned to the original IP.

6.3.3 EXAMPLES

Delete a node test_02 from the cluster.

```
Cluster> del test_02
```

Delete 2 nodes test_02,test_03 from the cluster.

```
Cluster> del test_02,test_03
```

6.3.4 SEE ALSO

add(1), reboot(1), show(1), shutdown(1)

6.4 reboot

6.4.1 SYNOPSIS

`reboot` *nodename*

6.4.2 DESCRIPTION

The cluster `reboot` command reboots a node or all of the nodes in the cluster. To reboot a node in the cluster, specify the *nodename* as it is displayed in the `cluster show` command. To reboot all of the nodes in the cluster, use `all` for *nodename*.

6.4.3 OPTIONS

nodename Node on which the operation takes place. A value of `all` indicates the operation takes place on all nodes of the cluster.

`reboot` *nodename* Reboots a node or all of the nodes in the cluster. To reboot a node in the cluster, specify the *nodename* as it is displayed in the `cluster show` command. To reboot all of the nodes in the cluster, use `all` for *nodename*.

6.4.4 EXAMPLES

Reboots a node in the cluster.

```
Cluster> reboot test_4
Stopping Cluster processes on test_4
Sent reboot command to test_4. SSH sessions to test_4 may terminate.
```

Reboots all of the nodes in the cluster.

```
Cluster> reboot all
Stopping Cluster processes on all
SSH sessions to all nodes may terminate.
Sent reboot command to test_2
Sent reboot command to test_3
Sent reboot command to test_4
Sent reboot command to test_1
```

6.4.5 SEE ALSO

show(1), shutdown(1)

6.5 show

6.5.1 SYNOPSIS

`show` [currentload]

6.5.2 DESCRIPTION

The cluster `show` command displays the nodes in the cluster, their states, CPU load, and network load during the past 15 minutes. It also displays information about nodes that are being added to the cluster, deleted from the cluster and nodes on which upgrade is in progress. The `rx` and `tx` columns display statistics of received and transmitted bytes respectively.

6.5.3 OPTIONS

`show` [*currentload*] If you specify currentload, the load statistics show the CPU load, and network load for the 5 seconds after the command is issued.

6.5.4 EXAMPLES

Display the current state of all of the nodes in the cluster and their loads during the past 15 minutes.

```
Cluster> show

Node            State      CPU(15 min)    pubeth0(15 min)    pubeth1(15 min)
                                 %        rx(MB/s) tx(MB/s)  rx(MB/s) tx(MB/s)
----            -----      -----------   -------- --------  -------- --------
test_1          RUNNING          1.35       0.00     0.00      0.00     0.00
test_2          RUNNING          1.96       0.00     0.00      0.00     0.00
test_3          FAULTED

Nodes in Transition
Node/IP                    Operation           State      Description
-------                    ---------           -----      -----------
10.200.58.202              Add node            FAILED     Installing packages
test_6                     Delete node         ONGOING    Removing node
test_4,test_5              Rolling upgrade     ONGOING    Rolling upgrade phase 2
```

Show the CPU and network loads collected for the next 5 seconds.

```
Cluster> show currentload

Node            State      CPU(5 sec)     pubeth0(5 sec)     pubeth1(5 sec)
                                 %        rx(MB/s) tx(MB/s)  rx(MB/s) tx(MB/s)
----            -----      -----------   -------- --------  -------- --------
test_1          RUNNING          1.35       0.00     0.00      0.00     0.00
test_2          RUNNING          1.96       0.00     0.00      0.00     0.00
test_3          FAULTED
```

6.5.5 SEE ALSO

reboot(1), shutdown(1)

6.6 shutdown

6.6.1 SYNOPSIS

`shutdown` *nodename*

6.6.2 DESCRIPTION

The cluster `shutdown` command shuts down the nodes in the cluster.

6.6.3 OPTIONS

nodename Node on which the operation takes place. A value of `all` indicates the operation takes place on all of the nodes in the cluster.

`shutdown` ***nodename*** Shut down a node or all of the nodes in the cluster. To shut down a node in the cluster, specify the *nodename*, as it appears in the `cluster show` command. To shut down all of the nodes in the cluster, use `all` for *nodename*.

6.6.4 EXAMPLES

Shut down a node in the cluster.

```
Cluster> shutdown test_4
Stopping Cluster processes on test_4
Sent shutdown command to test_4. SSH sessions to test_4 may terminate.
```

Shut down all of the nodes in the cluster.

```
Cluster> shutdown all
Stopping Cluster processes on all
SSH sessions to all nodes may terminate.
Sent shutdown command to test_2
Sent shutdown command to test_3
Sent shutdown command to test_4
Sent shutdown command to test_1
```

6.6.5 SEE ALSO

reboot(1), show(1)

7 DB Commands

7.1 database

7.1.1 SYNOPSIS

`fs create` *obj-type db-name fs-name size*

`fs destroy` *db-name fs-name*

`fs list`

`pool create` *obj-type disk1* [, *disk2*,...]

`pool destroy` *obj-type*

`pool list`

`share add` *obj-type export-dir* [*client*]

`share delete` *export-dir* [*client*]

`share show`

7.1.2 DESCRIPTION

These `database` commands are used to create and manage storage for Oracle database clients. The storage is accessed by Oracle hosts using the Oracle Direct NFS protocol. Oracle Direct NFS (DNFS) is an optimized NFS (Network File System) client that provides faster access to NFS storage located on NAS appliances. The Oracle Database Direct NFS client integrates the NFS client functionality directly in the Oracle database. Through this integration, the I/O path between Oracle and the NFS server is optimized, providing significantly better performance. The Oracle Direct NFS client outperforms traditional NFS clients, and is simple to configure, and it provides a standard NFS client implementation across all hardware and operating system platforms.

It is recommended to group the storage according to the database objects stored in the file system. Oracle database objects are broadly divided into REDO transaction logs, archived logs, table data, index, and tempfiles. The following values for ‘‘ obj-type‘‘ are available: txnlog : Stores REDO transaction logs data : Stores TABLE data of datafiles index : Stores INDEX data temp : Stores TEMPORARY files archivelog : Stores archive logs

7.1.3 OPTIONS

`fs create` ***obj-type db-name fs-name size*** Create a file system for database with name *db-name* to store objects of type *obj-type*. The name of the file system is *fs-name* and the size of the file system is *size*.

`fs destroy` ***db-name fs-name*** Destroy the file system with name *fs-name*.

`fs list` List file systems created for storing database files.

`pool create` ***obj-type disk1*** **[*, disk2,...*]** Create a pool to store database objects of type *obj-type1*. The *obj-type* must be one of txnlog, archivelog, data, index, or temp.

`pool destroy` ***obj-type*** Destroy the pool specified by the database *obj-type*.

`pool list` List all the pools configured for the database. `share add` *obj-type export-dir* [*client*] Share and export the file system *export-dir*. After this command, database clients are able to mount using NFS this file system on their host.

`share delete` ***export-dir*** **[*client*]** Delete (or unshare) the exported file system.

`share show` Display all the shared database file systems.

7.1.4 EXAMPLES

Here is an example of the commands to create storage for an Oracle TPCC database.

1. First create storage pools for the TPCC database.

   database> pool create data vmdk0_1,vmdk0_2 database> pool create txnlog vmdk0_3 database> pool create archivelog vmdk0_4 database> pool create index vmdk0_5,vmdk0_6

2. Next create the respective file systems.

   database> fs create data tpcc data1 300g database> fs create txnlog tpcc redo1 12g database> fs create index tpcc indx1 48g database> fs create index tpcc arch1 100g

3. Next share and export the file systems.

   database> share add txnlog /vx/tpcc_redo1 database> share add data /vx/tpcc_data1 database> share add index /vx/tpcc_indx1 database> share add archivelog /vx/tpcc_arch1

7.1.5 SEE ALSO

pool(1), fs(1), share(1)

7.2 fs

7.2.1 SYNOPSIS

`fs create` *obj-type db-name fs-name size*

`fs destroy` *db-name fs-name*

`fs list`

7.2.2 DESCRIPTION

These `database` commands are used to create and manage file systems for Oracle database clients. These file systems are accessed by Oracle hosts using the Oracle Direct NFS (DNFS) protocol. Oracle Direct NFS (DNFS) is an optimized NFS (Network File System) client that provides faster access to NFS storage located on NAS appliances.

It is recommended to group the file systems according to the database objects stored in the file systems. Oracle database objects are broadly divided into REDO transaction logs, archived logs, table data, index, and tempfiles. The following values for `obj-type` are available:

txnlog: Stores REDO transaction logs

data: Stores TABLE data of datafiles

index: Stores INDEX data

temp: Stores TEMPORARY files

archivelog: Stores archive logs

7.2.3 OPTIONS

`fs create` ***obj-type db-name fs-name size*** Create a file system for database with name *db-name* to store objects of type *obj-type*. The name of the file system is *fs-name* and the size of the file system is *size*.

`fs destroy` ***db-name fs-name*** Destroy the file system with name *fs-name*.

`fs list` List file systems created for storing database files.

7.2.4 EXAMPLES

Here is an example of the commands to create storage for an Oracle TPCC database.

1. Create file systems for the TPCC database. The following examples create four separate file systems to store respective database objects:

```
database> fs create data tpcc data1 300g
database> fs create txnlog tpcc redo1 12g
database> fs create index tpcc indx1 48g
database> fs create index tpcc arch1 100g
```

2. Display database file systems:

```
database> fs list
```

7.2.5 SEE ALSO

pool(1), fs(1), share(1)

7.3 pool

7.3.1 SYNOPSIS

`pool create` *obj-type disk1* [, *disk2*,...]

`pool destroy` *obj-type*

`pool list`

7.3.2 DESCRIPTION

These `database pool` commands are used to create and manage storage pools for an Oracle database. It is recommended to group the storage according to the database objects stored in the file system. Oracle database objects are broadly divided into REDO transaction logs, archived logs, table data, index, and tempfiles. The following values for `obj-type` are available.

txnlog : Stores REDO transaction logs

data : Stores TABLE data of datafiles

index : Stores INDEX data

temp : Stores TEMPORARY files

archivelog : Stores archive logs

7.3.3 OPTIONS

`pool create` *obj-type disk1* [, *disk2* ,...] Create a pool to store database objects of type *obj-type*. The *obj-type* must be one of txnlog, archivelog, data, index, or temp.

`pool destroy` *obj-type* Destroy the pool specified by the database *obj-type*.

`pool list` List all the pools configured for the database.

7.3.4 EXAMPLES

Create storage pools for the Oracle database.

1. First create storage pools for the Oracle database.

```
database> pool create data vmdk0_1,vmdk0_2
database> pool create txnlog vmdk0_3
database> pool create archivelog vmdk0_4
database> pool create index vmdk0_5,vmdk0_6
```

2. `database> pool list`

```
POOL NAME                       DISKS
---------                       -----
ora_DATA_pool                   vmdk0_1 vmdk0_2
ora_TXNLOG_pool                 vmdk0_3
ora_ARCHIVE_pool                vmdk0_4
ora_INDEX_pool                  vmdk0_5,vmdk0_6
```

7.3.5 SEE ALSO

pool(1), fs(1), share(1)

7.4 share

7.4.1 SYNOPSIS

`share add` *obj-type export-dir* [*client*]

`share delete` *export-dir* [*client*]

`share show`

7.4.2 DESCRIPTION

These `database` commands are used to export file systems created for databases using NFS. The Oracle Database Direct NFS client integrates the NFS client functionality directly in the Oracle database. Through this integration, the I/O path between Oracle and the NFS server is optimized, providing significantly better performance. The Oracle Direct NFS client outperforms traditional NFS clients, and is simple to configure, and it provides a standard NFS client implementation across all hardware and operating system platforms.

7.4.3 OPTIONS

`share add` *obj-type export-dir* [*client*] Share and export the file system *export-dir*. After this command, database clients are able to mount using NFS this file system on their host.

`share delete` *export-dir* [*client*] Delete (or unshare) the exported file system.

`share show` Display all the shared database file systems.

7.4.4 EXAMPLES

Here is an example of commands to export file systems for an Oracle TPCC database.

1. Share and export the file systems for the TPCC database.

```
database> share add txnlog /vx/tpcc_redo1
database> share add data /vx/tpcc_data1
database> share add index /vx/tpcc_indx1
database> share add archivelog /vx/tpcc_arch1
```

2. Show the exported file systems for the database.

```
database> share show
```

7.4.5 SEE ALSO

pool(1), fs(1), share(1)

8 FTP Commands

8.1 FTP

8.1.1 SYNOPSIS

`server` start|stop|status

`set listen_port` *port_number*

`set listen_ipv6` yes|no

`set max_connections` *connections_number*

`set max_conn_per_client` *connections_number*

`set passive_port_range` *port_range*

`set idle_timeout` *time_in_minutes*

`set allow_non_ssl` yes|no

`set umask` *file_umask*

`set anonymous_logon` yes|no

`set anonymous_login_dir` *login_directory*

`set anonymous_write` yes|no

`set user_logon` yes|no

`set homedir_path` *home_directory_path*

`set allow_delete` yes|no

`set security` nis_ldap|ads|local

`set chroot_users` yes|no

`set create_homedirs` yes|no

`unset anonymous_login_dir`

`unset homedir_path`

`show`

`session show`

`session showdetail` [*filter_options*]

`session terminate` *session_id*

`logupload` *url* [*nodename*]

`local user add` *username*

`local user passwd` *username*

`local user delete` *username*

`local user show`

`local user set bandwidth` *username bandwidth*

`local user set homedir` *username homedir*

8.1.2 DESCRIPTION

The `FTP` commands are used to configure the FTP server. The FTP server allows files to be accessed using the File Transfer Protocol (FTP).

By default, the FTP server is not running. To start the FTP server, enter the `server start` command.

By default, anonymous FTP access is disabled. To enable anonymous FTP access, use `set anonymous_logon` *yes*. The UID of anonymous user is 40, GID is 49, and username is 'ftp'. To change the default login directory of anonymous users, use `set anonymous_login_dir`. By default, anonymous users do not have write access to their login directory. To allow anonymous users write access to their login directories, use `set anonymous_write` *yes*.

By default, the FTP server allows plain text and SSL (secure) user logins. To allow only SSL (secure) and to reject plain text user logins, use `set allow_non_ssl` *no*.

If the `create_homedirs` option is set to `yes`, Veritas Access creates a user's home directory on the FTP server with the same name that was retrieved from the authentication server. This directory is used internally. If the `create_homedirs` option is set to `no`, the Veritas Access administrator must manually create a directory that matches the home directory on the authentication server. Regardless of the setting of the `create_homedirs` option, the Veritas Access administrator must manually create the user's directory where the user logs in. This directory is in the location specified by the `homedir_path` option. The directory must have execute permissions set. The following examples show the functionality for different types of users, if the `homedir_path` is set to `/vx/fs1/ftphomes` and `create_homedirs` option is set to `yes`.

For NIS users: Create the directory with username in the homedir_path. For example: `/vx/fs1/ftphomes/nisuser1`.

For LDAP users: Create the directory with ldapusername in the homedir_path. For example: `/vx/fs1/ftphomes/ldapuser1`.

For AD users: The home directory on the AD server is specified as domainname\username. The full string (domainname\username) identifies the AD user. Create the directory domainname\username in the homedir_path. For example: `/vx/fs1/ftphomes/mydomain\\aduser1`.

For local users: If the homedir_path is set to /vx/fs1/ftphomes, Veritas Access creates the directory `/vx/fs1/ftphomes/username`.

If `create_homedirs` is set to *no* and user's home directory does not exist or has not been created manually, then the login fails. If user's home directory has been manually created, then the administrator has to ensure that the permissions have been set correctly.

If `chroot_users` is set to *yes*, then users are restricted to their home directories. If `homedir_path` is set to /vx/fs1/ftphomes, `chroot_users` is set to *yes*, and an AD user DOMAIN\aduser1 logs in using FTP, then the user is restricted to /vx/fs1/ftphomes/home/DOMAIN/adsuser1. If `chroot_users` is set to *no* for the above case, then the user is restricted to /vx/fs1/ftphomes.

For *security local*, the `chroot_users` value should be *yes*.

The `umask` defines the mask for permissions with which files or directories are created using FTP. If *file_umask* is set to 177, then new files are created as having permissions 600, which defines rw———. The owner of this file has read and write permissions to this file, members in the users group do not have read/write permissions.

The `server stop` command terminates any existing FTP sessions.

8.1.3 OPTIONS

bandwidth Upload or download bandwidth in Bytes/sec.

nodename Node on which the operation takes place. A value of `all` indicates that the operation takes place on all nodes in the cluster.

connections_number This is an integer, between 1 to 9999. This value defines the maximum number of simultaneous connections allowed for each node.

port_range The *port_range* specifies a range of ports represented as startingport:endingport, both inclusive. The value of port numbers used should range from 30000 to 50000.

time_in_minutes The value of *time_in_minutes* should range from 1 to 600. The *time_in_minutes* defines the amount of idle time after which a session is disconnected.

login_directory The directory into which anonymous users login. The value of *login_directory* starts with /vx/.

home_directory_path The directory in which user's directory should be created manually. The value of *home_directory_path* starts with /vx/.

session_id The unique identifier for each FTP session. It is the value that is shown in `session showdetail`.

filter_options Filters the sessions shown in `session showdetail`. Filter options include client_ip, server_ip, and user. Filter options can be combined by using ','. If multiple filter options are used, sessions matching all the filter options are shown.

`server` start|stop|status Start, stop, or display the status of the FTP server. All configuration changes made using the `set` command come into effect only when the server is restarted. For example, `set` command can be used to set the max_connections to 9999, but changes take effect only when the FTP server is restarted using `server stop` followed by `server start`. `server stop` terminates all existing FTP connections.

`set listen_port` *port_number* This option specifies the port number on which the FTP service should listen for connections. Valid values for this parameter range from 10 to 1023. The default value of this parameter is 21. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

`set listen_ipv6` yes|no This option specifies if the FTP service should listen on IPv6 for connections. The default value of this parameter is no. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

`set max_connections` *connections_number* This command specifies the maximum number of simultaneous FTP clients allowed per node. The default value of this parameter is 2000. Valid values for this parameter range from 1 to 9999. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

set max_conn_per_client ***connections_number*** This option specifies the maximum number of simultaneous FTP connections allowed per client IP per server node. Valid values for this parameter range from 1 to 9999. The default value of this parameter is 2000. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

set passive_port_range ***port_range*** This command specifies the range of port numbers to listen on for passive FTP transfers. The *port_range* defines a range specified as startingport:endingport. For example, a *port_range* of 30000:40000 specifies that port numbers starting from 30000 to 40000 can be used for passive FTP. Valid values for port numbers range from 30000 to 50000. The default value of this option is 30000:40000. These changes take effect when the FTP server is restarted using `server stop` followed by `server start`.

set idle_timeout ***time_in_minutes*** This command specifies the amount of time in minutes after which an idle connection is disconnected. Valid values for *time_in_minutes* range from 1 to 600. Default value for *time_in_minutes* is 15 minutes. These changes take effect when the FTP server is restarted using `server stop` followed by `server start`.

set allow_non_ssl yes|no This command specifies whether or not to allow non-secure (plain-text) logins into the FTP server. Entering *yes* allows non-secure (plain-text) logins to succeed. Entering *no* causes non-secure (plain-text) logins to fail. The default value for this parameter is *yes*. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

set umask ***file_umask*** The `umask` defines the mask for permissions with which files or directories are created using FTP. If *file_umask* is set to 177, then new files are created as having permissions 600, which defines rw———. The owner of this file has read and write permissions to this file, members in the users group do not have read/write permissions. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

set anonymous_logon yes|no This tells the FTP server whether or not to allow anonymous logins to the FTP server. Entering *yes* allows anonymous access to the FTP server. Entering *no* does not allow anonymous access. The default value of this parameter is *no*. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

set anonymous_login_dir ***login_directory*** This command specifies the login directory of anonymous users. Valid values of this parameter start with */vx/*. The administrator must ensure that the anonymous user (UID:40 GID:49 UNAME:ftp) has appropriate permissions to read files in *login_directory*. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

set anonymous_write yes|no This command specifies whether or not anonymous users have write access in their login directory. Enter *yes* to allow anonymous users to modify contents of their *login_directory*. A value of *no* does not allow anonymous users to modify contents of their *login_directory*. The administrator must ensure that anonymous user (UID:40 GID:49 UNAME:ftp) has the appropriate permissions to modify files in *login_directory*. The default value of this parameter is *no*. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

set user_logon yes|no This command specifies whether to allow FTP access for users. A value of *yes* would allow normal users (non-anonymous) to login. The default value of this parameter is *yes*. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

set homedir_path ***login_directory*** This option specifies the login directory of normal users. Valid values of this parameter start with */vx/*. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

set allow_delete yes|no This option specifies whether users are allowed to delete files on the FTP server. A value of *yes* allows users to delete uploaded files. This option only affects users and does

not affect anonymous logins. Anonymous logins are never allowed to delete files. The default value of this parameter is *yes*. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

`set security` nis_ldap|ads|local This option specifies what kind of users are allowed to login to the FTP server. A value of *nis_ldap* allows users configured on NIS or an LDAP server to login to the FTP server, and users created using `local user` or from Windows active directory are not allowed to login. A value of *ads* allows users configured on Windows active directory as specified in `cifs show`, NIS/LDAP, and `local user` users are not allowed to login. A value of *local* allows users created using `local user` to login to the FTP server, and NIS/LDAP/AD users are not allowed to login. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

`set chroot_users` yes|no This options specifies whether users should be restricted to their home directories. A value of *yes* limits users to his home directory. A value of *no* allows users to view files in parent directories, user is still restricted in `homedir_path`. If `security` is *local*, then `chroot_users` should be *yes*. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

`set create_homedirs` yes|no This option specifies if home directories should be created when a user logs in, if it does not exist. A value of *yes* allows the FTP server to create the user's home directory, if it does not exist. If this is set to *no*, then there should already be a home directory created for this user, and the user should have permissions to read/execute in this directory, otherwise the login fails. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

`unset anonymous_login_dir` This option sets the login directory of anonymous users to the default '-'. Changes take effect when the FTP server is restarted using `server stop` followed by `server start`.

`unset homedir_path` This option sets the login directory of normal users to the default '-'. Changes take effect when the FTP server is restarted using `server stop` followed by `server start`.

`show` Displays the list of all configurable options and their values. Use the `set` command to modify these values.

`session show` Displays the number of current FTP sessions for each node.

`session showdetail` [*filter_options*] Displays the details of the current FTP sessions.

`session terminate` *session_id* Terminates the FTP session using the session id.

`logupload` *url* [*nodename*] Uploads FTP logs to the URL *url*. The *url* supports FTP and SCP. If the *nodename* is specified, only logs from the *nodename* are uploaded. The default name for the uploaded file is ftp_log.tar.gz. Passwords added directly to the *url* are not supported.

`local user add` *username* Add a local FTP user with name *username*. This user gets authenticated when security is set to local. This command prompts for a password to be set for this user. The default home-directory for a user is the same as the *username*. Users created using this command can only access the files in their home directory and cannot see files of other users. Username can contain up to 32 characters.

`local user passwd` *username* Change password for the local FTP user with name *username*.

`local user delete` *username* Delete the *username* of the FTP user. This does not delete the *username*'s home directory. Home directory for *username* still continues to exist after deletion of the user.

`local user show` Show the list of users and information associated with the users.

local user set bandwidth ***username bandwidth*** Set the bandwidth for *username* to *bandwidth*. By default, there is no limit on the upload/download bandwidth limit for users. *bandwidth* is specified in Bytes/sec. Changes to this value are applicable for any new connections. If *bandwidth* is *0*, then `bandwidth` is reset to unlimited bandwidth.

local user set homedir ***username homedir*** Set the home directory for *username* to *homedir*. This does not migrate any existing data from *username*'s current home directory to *homedir*. *homedir* is relative to *homedir_path* that is set using *set homedir_path*. Changes to this value are applicable for any new connections.

8.1.4 EXAMPLES

Start the FTP server.

```
FTP> server start
```

View the current status of the FTP server.

```
    FTP> server status
FTP Status on node_1 : ONLINE
FTP Status on node_2 : ONLINE
```

Stop the FTP server. (Terminates any existing FTP sessions.)

```
FTP> server stop
```

Display the current FTP server configuration.

```
    FTP> show
Parameter                Current Value        New Value
---------                -------------        ---------
listen_port              21
listen_ipv6              no
max_connections          2000
max_conn_per_client      1000
passive_port_range       30000:40000
allow_non_ssl            yes
idle_timeout             15 minutes
umask                    177
anonymous_logon          no                   yes
anonymous_write          no
anonymous_login_dir      /vx/ftpanon
user_logon               yes
homedir_path             /vx/ftphomes
allow_delete             yes
security                 local
chroot_users             yes
create_homedirs          yes
```

Change the maximum number of concurrent FTP connections allowed on the FTP server (restarting the FTP server will bring the changes into effect).

```
    FTP> set max_connections 3000
Changes would be applicable after restart of FTP service.
Success.

FTP> show
```

```
Parameter                Current Value        New Value
---------                -------------        ---------
listen_port              21
listen_ipv6              no
max_connections          2000                 3000
max_conn_per_client      1000
passive_port_range       30000:40000
allow_non_ssl            yes
idle_timeout             15 minutes
umask                    177
anonymous_logon          yes
anonymous_write          yes
anonymous_login_dir      /vx/ftp_fs/anon_dir
user_logon               no
homedir_path             /vx/ftp_fs/users_dir
allow_delete             yes
security                 local
chroot_users             yes
create_homedirs          yes

FTP> server start
FTP> show
Parameter                Current Value
---------                -------------
listen_port              21
listen_ipv6              no
max_connections          3000
max_conn_per_client      1000
passive_port_range       30000:40000
allow_non_ssl            yes
idle_timeout             15 minutes
umask                    177
anonymous_logon          yes
anonymous_write          yes
anonymous_login_dir      /vx/ftp_fs/anon_dir
user_logon               no
homedir_path             /vx/ftp_fs/users_dir
allow_delete             yes
security                 local
chroot_users             yes
create_homedirs          yes
```

Display the number of current FTP sessions.

```
    FTP> session show
Max Sessions : 2000

Nodename             Current Sessions
--------             ----------------
node_1                4
node_2                2
```

To view the details of all the current FTP sessions.

```
    FTP> session showdetail
Session ID       User         Client IP       Server IP       State  File
----------       ----         ---------       ---------       -----  ----
node_1.1111       user1        10.209.105.219  10.209.105.111  IDLE
```

```
node_1.1112       user2        10.209.106.11   10.209.105.111  IDLE
node_2.1113       user3        10.209.107.21   10.209.105.112  IDLE
node_1.1117       user4        10.209.105.219  10.209.105.111  RETR  file123
node_2.1118       user1        10.209.105.219  10.209.105.111  STOR  file345
node_1.1121       user5        10.209.111.219  10.209.105.112  IDLE
```

Upload the logs from all of the nodes to an SCP-based URL.

```
    FTP> logupload scp://user@host:/path/to/directory
Password:
Collecting FTP logs, please wait.....
Uploading the logs to scp://user@host:/path/to/directory, please wait...done
```

Upload the logs from node_1 to an FTP-based URL.

```
    FTP> logupload ftp://user@host:/path/to/directory node_1
Password:
Collecting FTP logs, please wait.....
Uploading the logs to ftp://user@host:/path/to/directory, please wait...done
```

8.1.5 SEE ALSO

server(1), show(1), set(1), session(1), logupload(1), local(1), unset(1)

8.2 local

8.2.1 SYNOPSIS

`local user add` *username*

`local user passwd` *username*

`local user delete` *username*

`local user show`

`local user set bandwidth` *username bandwidth*

`local user set homedir` *username homedir*

8.2.2 DESCRIPTION

The `local` commands add, delete, or display FTP users. Local users' home directories are automatically created when they login for the first time.

8.2.3 OPTIONS

bandwidth Upload or download bandwidth in Bytes/sec.

`local user add` ***username*** Add a local FTP user with name *username*. This user gets authenticated when security is set to local. This command prompts for a password to be set for this user. The default home directory for a user is the same as the *username*. Users created using this command can only access the files in their home directory and cannot see files of other users. Username can contain up to 32 characters.

`local user passwd` ***username*** Change password for the local FTP user with name *username*.

`local user delete` ***username*** Delete the *username* FTP user. This does not delete the *username*'s home directory. Home directory for *username* still continues to exist after deletion of the user.

`local user show` Show the list of users and information associated with the users.

`local user set bandwidth` ***username bandwidth*** Set the bandwidth for *username* to *bandwidth*. By default, there is no limit on upload/download bandwidth limit for users. *bandwidth* is specified in Bytes/sec. Changes to this value are applicable for any new connections. If *bandwidth* is *0*, then `download_bandwidth` is reset to unlimited bandwidth.

`local user set homedir` ***username homedir*** Set the home directory for *username* to *homedir*. This does not migrate any existing data from the *username*'s current home directory to *homedir*. *homedir* is relative to *homedir_path* that is set using *set homedir_path*. Changes to this value are applicable for any new connections.

8.2.4 EXAMPLES

Adds a local FTP user `localuser1`.

```
FTP> local user add localuser1
Input password for localuser1.
Enter password:
```

```
Re-enter password:
Success.
```

Changes the password for user `localuser1`.

```
FTP> local user passwd localuser1
Input password for localuser1.
Enter password:
Re-enter password:
Success.
```

Deletes user `localuser1`.

```
FTP> local user delete localuser1
Success.
```

Show the list of local FTP users.

```
FTP> local user show
USER         HOMEDIR          BANDWIDTH
----         -------          --------
localuser1   /localuser1           -
localuser2   /localuser2        2046
localuser3   /localuser3           -
localuser4   /localuser4           -
```

Changes the home directory of *localuser4* /localuser4_new.

```
FTP> local user set homedir localuser3 /localuser4_new
Success.

FTP> local user show
USER         HOMEDIR          BANDWIDTH
----         -------          --------
localuser1   /localuser1           -
localuser2   /localuser2        2046
localuser3   /localuser3           -
localuser4   /localuser4_new       -
```

8.2.5 SEE ALSO

server(1), set(1)

8.3 logupload

8.3.1 SYNOPSIS

`logupload` *url* [*nodename*]

8.3.2 DESCRIPTION

The `logupload` command allows the user to upload the FTP server logs to *url*. The default name for the uploaded file is ftp_log.tar.gz.

8.3.3 OPTIONS

nodename The node on which the operation occurs. A value of `all` indicates the operation takes place on all of the nodes in the cluster.

`logupload` *url* [*nodename*] Uploads FTP logs to the URL *url*. The *url* supports FTP and SCP. If *nodename* is specified, only logs from the node *nodename* are uploaded. The default name for the uploaded file is ftp_log.tar.gz. Passwords added directly to the *url* are not supported.

8.3.4 EXAMPLES

To upload the logs from all of the nodes to an SCP-based URL.

```
    FTP> logupload scp://user@host:/path/to/directory
Password:
Collecting FTP logs, please wait.....
Uploading the logs to scp://user@host:/path/to/directory, please wait...done
```

To upload the logs from node_1 to an FTP-based URL.

```
    FTP> logupload ftp://user@host:/path/to/directory node_1
Password:
Collecting FTP logs, please wait.....
Uploading the logs to ftp://user@host:/path/to/directory, please wait...done
```

8.3.5 SEE ALSO

server(1), show(1), set(1), session(1)

8.4 server

8.4.1 SYNOPSIS

`server` [start|stop|status]

8.4.2 DESCRIPTION

The FTP `server` command starts, stops, or checks the status of the FTP server. The `server stop` command terminates any existing FTP sessions.

8.4.3 OPTIONS

`server` [start|stop|status] Start, stop, or check the status of the FTP server.

8.4.4 EXAMPLES

Start the FTP server. If the FTP server is already started, Veritas Access clears the faults (if any), and then tries to start the FTP server.

```
    FTP> server start
Success.
FTP> server status
FTP Status on node_1 : ONLINE
FTP Status on node_2 : ONLINE
```

Stop the FTP server. Terminates any existing FTP sessions.

```
    FTP> server stop
Success.
FTP> server status
FTP Status on node_1 : OFFLINE
FTP Status on node_2 : OFFLINE
```

8.4.5 SEE ALSO

set(1), show(1), session(1), logupload(1)

8.5 session

8.5.1 SYNOPSIS

`session show`

`session showdetail` [*filter_options*]

`session terminate` *session_id*

8.5.2 DESCRIPTION

The `session` commands allow the user to view or terminate the current FTP sessions to the server.

8.5.3 OPTIONS

session_id The unique identifier for each FTP session. It is the value that is displayed in `session showdetail`.

filter_options The *filter_options* filter the sessions displayed in `session showdetail`. Filter options include client_ip, server_ip, and user. Filter options can be combined by using ','. If multiple filter options are used, sessions matching all of the filter options are displayed.

`session show` Displays the number of current FTP sessions on each node.

`session showdetail` [*filter_options*] Displays the details of each session that matches the *filter_options* criteria. If no *filter_options* is specified, all the sessions are displayed. If multiple filter options are provided, then sessions matching all the filter options are displayed. For each session, user, client IP, server IP, session state (Uploading/Downloading/Idle), file (uploading/downloading file) is displayed. A value of '?' for the user signifies the session is not authenticated yet.

`session terminate` *session_id* Terminates the session with the session ID as *session_id*. The *session_id* is the value displayed in `session showdetail`.

8.5.4 EXAMPLES

To view the number of current FTP sessions.

```
    FTP> session show
Max Sessions : 2000
Nodename             Current Sessions
--------             ----------------
node_1                4
node_2                2
```

To view the details of all the current FTP sessions.

```
    FTP> session showdetail
Session ID       User         Client IP       Server IP       State  File
----------       ----         ---------       ---------       -----  ----
node_1.1111       user1        10.209.105.219  10.209.105.111  IDLE
node_1.1112       user2        10.209.106.11   10.209.105.111  IDLE
node_2.1113       user3        10.209.107.21   10.209.105.112  IDLE
node_1.1117       user4        10.209.105.219  10.209.105.111  RETR  file123
```

```
node_2.1118       user1        10.209.105.219  10.209.105.111  STOR  file345
node_1.1121       user5        10.209.111.219  10.209.105.112  IDLE
```

To view the details of the current FTP sessions originating from client 10.209.107.21.

```
    FTP> session showdetail client_ip=10.209.107.21
Session ID       User         Client IP       Server IP       State  File
----------       ----         ---------       ---------       -----  ----
node_2.1113       user3        10.209.107.21   10.209.105.112  IDLE
```

To view the details of the current FTP sessions to the server IP 10.209.105.112.

```
    FTP> session showdetail server_ip=10.209.105.112
Session ID       User         Client IP       Server IP       State  File
----------       ----         ---------       ---------       -----  ----
node_2.1113       user3        10.209.107.21   10.209.105.112  IDLE
node_1.1121       user5        10.209.111.219  10.209.105.112  IDLE
```

To view the details of the current FTP sessions to the server IP 10.209.105.112 originating from client 10.209.107.21.

```
    FTP> session showdetail server_ip=10.209.105.112,client_ip=10.209.107.21
Session ID       User         Client IP       Server IP       State  File
----------       ----         ---------       ---------       -----  ----
node_2.1113       user3        10.209.107.21   10.209.105.112  IDLE
```

To view the details of the current FTP sessions to the server IP 10.209.105.112 by user user3.

```
    FTP> session showdetail server_ip=10.209.105.112,user=user3
Session ID       User         Client IP       Server IP       State  File
----------       ----         ---------       ---------       -----  ----
node_2.1113       user3        10.209.107.21   10.209.105.112  IDLE
```

To terminate a FTP session that is displayed in `session showdetail`.

```
    FTP> session terminate node_2.1113

Session node_2.1113 terminated
```

8.5.5 SEE ALSO

server(1), show(1), set(1), logupload(1)

8.6 set

8.6.1 SYNOPSIS

`set listen_port` *port_number*

`set listen_ipv6` yes|no

`set max_connections` *connections_number*

`set max_conn_per_client` *connections_number*

`set passive_port_range` *port_range*

`set idle_timeout` *time_in_minutes*

`set allow_non_ssl` yes|no

`set umask` *file_umask*

`set anonymous_logon` yes|no `set anonymous_login_dir` *login_directory* `set anonymous_write` yes|no

`set user_logon` yes|no `set homedir_path` *home_directory_path* `set allow_delete` yes|no `set security` nis_ldapladsllocal `set chroot_users` yes|no `set create_homedirs` yes|no

8.6.2 DESCRIPTION

The `set` commands allow the user to set various configurable options for the FTP server.

8.6.3 OPTIONS

login_directory Login directory is the directory into which users login. The value of *login_directory* starts with /vx.

home_directory_path The directory in which users home directories are created. The value of *home_directory_path* starts with /vx/.

connections_number This is an integer, with a range between 1 to 9999. This value defines the maximum number of simultaneous connections allowed for each node.

port_range The *port_range* specifies a range of ports represented as startingport:endingport. Both ports are inclusive. The port numbers use a range from 30000 to 50000.

time_in_minutes Values for *time_in_minutes* range from 1 to 600. The *time_in_minutes* defines the amount of idle time after which a session is disconnected.

`set listen_port` *port_number* This option specifies the port number on which the FTP service should listen for connections. Valid values for this parameter range from 10 to 1023. The default value of this parameter is 21. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

`set listen_ipv6` yes|np This option specifies if the FTP service should listen on IPv6 connections. The default value of this parameter is no. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

`set max_connections` *connections_number* This command specifies the maximum number of simultaneous FTP clients allowed per node. The default value of this parameter is 2000. Valid values

for this parameter range from 1 to 9999. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

set max_conn_per_client ***connections_number*** This option specifies the maximum number of simultaneous FTP connections allowed per client IP per server node. Valid values for this parameter range from 1 to 9999. The default value of this parameter is 2000. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

set passive_port_range ***port_range*** This command specifies the range of port numbers to listen on for passive FTP transfers. The *port_range* defines a range specified as startingport:endingport. For example, a *port_range* of 30000:40000 specifies that port numbers starting from 30000 to 40000 can be used for passive FTP. Valid values for port numbers range from 30000 to 50000. The default value of this option is 30000:40000. These changes take effect when the FTP server is restarted using `server stop` followed by `server start`.

set idle_timeout ***time_in_minutes*** This command specifies the amount of time in minutes after which an idle connection is disconnected. Valid values for *time_in_minutes* range from 1 to 600. The default value for *time_in_minutes* is 15 minutes. These changes take effect when the FTP server is restarted using `server stop` followed by `server start`.

set allow_non_ssl yes|no This command specifies whether or not to allow non-secure (plain-text) logins into the FTP server. Entering *yes* allows non-secure (plain-text) logins to succeed. Entering *no* causes non-secure (plain-text) logins to fail. The default value for this parameter is *yes*. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

set umask ***file_umask*** The `umask` defines the mask for permissions with which files or directories are created using FTP. If *file_umask* is set to 177, then new files are created as having permissions 600, which defines rw———. The owner of this file has read and write permissions to this file, members in the users group do not have read/write permissions. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

set anonymous_logon yes|no This tells the FTP server whether or not to allow anonymous logins to the FTP server. Entering *yes* allows anonymous users to login to the FTP server. Entering *no* does not allow anonymous logins. The default value of this parameter is *no*. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

set anonymous_login_dir ***login_directory*** This command specifies the login directory of anonymous users. Valid values of this parameter start with */vx/*. The administrator must ensure that the anonymous user (UID:40 GID:49 UNAME:ftp) has the appropriate permissions to read files in *login_directory*. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

set anonymous_write yes|no This command specifies whether or not anonymous users can write to their login directory. Enter *yes* to allow anonymous users to modify contents of their *login_directory*. A value of *no* does not allow anonymous users to modify contents of their *login_directory*. The administrator must ensure that anonymous user (UID:40 GID:49 UNAME:ftp) has appropriate permissions to modify files in *login_directory*. The default value of this parameter is *no*. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

set user_logon yes|no This command specifies whether to allow FTP access for users. A value of *yes* allows normal users(non-anonymous) to login. The default value of this parameter is *yes*. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

set homedir_path ***login_directory*** This option specifies the login directory of normal users. Valid values of this parameter start with */vx/*. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

set allow_delete yes|no This option specifies whether users are allowed to delete files on the FTP server. A value of *yes* allows users to delete uploaded files. This option only affects users and does not affect anonymous logins. Anonymous logins are never allowed to delete files. The default value of this parameter is *yes*. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

set security nis_ldap|ads|local This option specifies what kind of users are allowed to login to the FTP server. A value of *nis_ldap* allows users configured on NIS or an LDAP server to login to the FTP server, and users created using `local user` or from Windows active directory are not allowed to login. A value of *ads* allows users configured on Windows active directory as specified in `cifs show`, NIS/LDAP and `local user` users are not allowed to login. A value of *local* allows users created using `local user` to login to the FTP server, and NIS/LDAP/AD users are not allowed to login. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

set chroot_users yes|no This options specifies whether users should be restricted to their home directories. A value of *yes* limits users to his home directory. A value of *no* will allow users to view files in parent directories, user will still be restricted of `homedir_path`. If `security` is *local*, then `chroot_users` should be *yes*. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

set create_homedirs yes|no This option specifies if home directories should be created when a user logs in, if it does not exist. A value of *yes* lets FTP create the user's home directory, if it does not exist. If this is set to *no*, then there should already be a home directory created for this user and the user should have permissions to read/execute in this directory, otherwise the login fails. Changes to this value take effect when the FTP server is restarted using `server stop` followed by `server start`.

8.6.4 EXAMPLES

Allows anonymous logins to the FTP server. Restart the FTP server to bring the changes into effect.

```
    FTP> set anonymous_logon yes
Changes would be applicable after the restart of the FTP service.
Success.

FTP> show
Parameter                Current Value        New Value
---------                -------------        ---------
listen_port              21
listen_ipv6              no
max_connections          2000
max_conn_per_client      1000
passive_port_range       30000:40000
idle_timeout             15 minutes
allow_non_ssl            yes
umask                    177
anonymous_logon          no                   yes
anonymous_write          no
anonymous_login_dir      /vx/ftpanon
user_logon               yes
homedir_path             /vx/ftphomes
allow_delete             yes
security                 local
chroot_users             yes
create_homedirs          yes
```

```
FTP> server stop
Success.
FTP> server start
Success.

FTP> show
Parameter                Current Value
---------                -------------
listen_port              21
listen_ipv6              no
max_connections          2000
max_conn_per_client      1000
passive_port_range       30000:40000
idle_timeout             15 minutes
allow_non_ssl            yes
umask                    177
anonymous_logon          yes
anonymous_write          no
anonymous_login_dir      /vx/ftpanon
user_logon               yes
homedir_path             /vx/ftphomes
allow_delete             yes
security                 local
chroot_users             yes
create_homedirs          yes
```

Allows anonymous users to have write permissions to their login directory on the FTP server. Restart the FTP server to bring the changes into effect.

```
    FTP> set anonymous_write yes

Changes would be applicable after restart of FTP service.
Success.

FTP> show
Parameter                Current Value        New Value
---------                -------------        ---------
listen_port              21
listen_ipv6              no
max_connections          2000
max_conn_per_client      1000
passive_port_range       30000:40000
idle_timeout             15 minutes
allow_non_ssl            yes
umask                    177
anonymous_logon          yes
anonymous_write          no                   yes
anonymous_login_dir      /vx/ftpanon
user_logon               yes
homedir_path             /vx/ftphomes
allow_delete             yes
security                 local
chroot_users             yes
create_homedirs          yes

FTP> server stop
Success.
```

```
FTP> server start
Success.

FTP> show
Parameter                Current Value
---------                -------------
listen_port              21
listen_ipv6              no
max_connections          2000
max_conn_per_client      1000
passive_port_range       30000:40000
idle_timeout             15 minutes
allow_non_ssl            yes
umask                    177
anonymous_logon          yes
anonymous_write          yes
anonymous_login_dir      /vx/ftpanon
user_logon               yes
homedir_path             /vx/ftphomes
allow_delete             yes
security                 local
chroot_users             yes
create_homedirs          yes
```

Changes the login directory of the anonymous user on the FTP server. Restart the FTP server to bring the changes into effect.

```
    FTP> set anonymous_login_dir /vx/ftp_fs/anon_dir

Changes would be applicable after restart of FTP service.
Success.

FTP> show
Parameter                Current Value        New Value
---------                -------------        ---------
listen_port              21
listen_ipv6              no
max_connections          2000
max_conn_per_client      1000
passive_port_range       30000:40000
idle_timeout             15 minutes
allow_non_ssl            yes
umask                    177
anonymous_logon          yes
anonymous_write          yes
anonymous_login_dir      /vx/ftpanon          /vx/ftp_fs/anon_dir
user_logon               yes
homedir_path             /vx/ftphomes
allow_delete             yes
security                 local
chroot_users             yes
create_homedirs          yes

FTP> server stop
Success.

FTP> server start
Success.
```

```
FTP> show
Parameter                Current Value
---------                -------------
listen_port              21
listen_ipv6              no
max_connections          2000
max_conn_per_client      1000
passive_port_range       30000:40000
idle_timeout             15 minutes
allow_non_ssl            yes
umask                    177
anonymous_logon          yes
anonymous_write          yes
anonymous_login_dir      /vx/ftp_fs/anon_dir
user_logon               yes
homedir_path             /vx/ftphomes
allow_delete             yes
security                 local
chroot_users             yes
create_homedirs          yes
```

Allows only anonymous access to the FTP server, user access is denied. Restart the FTP server to bring the changes into effect.

```
    FTP> set user_logon no

Changes would be applicable after restart of FTP service.
Success.

FTP> server start
Success.

FTP> show

Parameter                Current Value
---------                -------------
listen_port              21
listen_ipv6              no
max_connections          2000
max_conn_per_client      1000
passive_port_range       30000:40000
idle_timeout             15 minutes
allow_non_ssl            yes
umask                    177
anonymous_logon          yes
anonymous_write          yes
anonymous_login_dir      /vx/ftp_fs/anon_dir
user_logon               no
homedir_path             /vx/ftphomes
allow_delete             yes
security                 local
chroot_users             yes
create_homedirs          yes
```

Changes the login directory of the normal user (non-anonymous users) on the FTP server. Restart the FTP server to bring the changes into effect.

```
    FTP> set homedir_path /vx/ftp_fs/users_dir

Changes would be applicable after restart of FTP service.
Success.

FTP> server stop
Success.
FTP> server start
Success.

Parameter                Current Value
---------                -------------
listen_port              21
listen_ipv6              no
max_connections          2000
max_conn_per_client      1000
passive_port_range       30000:40000
idle_timeout             15 minutes
allow_non_ssl            yes
umask                    177
anonymous_logon          yes
anonymous_write          yes
anonymous_login_dir      /vx/ftp_fs/anon_dir
user_logon               no
homedir_path             /vx/ftp_fs/users_dir
allow_delete             yes
security                 local
chroot_users             yes
create_homedirs          yes
```

Changes the maximum number of concurrent FTP connections allowed on the FTP server. Restart the FTP server to bring the changes into effect.

```
    FTP> set max_connections 3000
Changes would be applicable after restart of FTP service.
Success.

FTP> server stop
Success.

FTP> server start
Success.

FTP> show
Parameter                Current Value
---------                -------------
listen_port              21
listen_ipv6              no
max_connections          3000
max_conn_per_client      1000
passive_port_range       30000:40000
idle_timeout             15 minutes
allow_non_ssl            yes
umask                    177
anonymous_logon          yes
anonymous_write          yes
anonymous_login_dir      /vx/ftp_fs/anon_dir
user_logon               no
```

```
homedir_path             /vx/ftp_fs/users_dir
allow_delete             yes
security                 local
chroot_users             yes
create_homedirs          yes
```

Changes the maximum number of concurrent FTP connections allowed to the FTP server from a client IP. Restart the FTP server to bring the changes into effect.

```
    FTP> set max_conn_per_client 500
Changes would be applicable after restart of FTP service.
Success.

FTP> server stop
Success.

FTP> server start
Success.

FTP> show
Parameter                Current Value
---------                -------------
listen_port              21
listen_ipv6              no
max_connections          3000
max_conn_per_client      500
passive_port_range       30000:40000
idle_timeout             15 minutes
allow_non_ssl            yes
umask                    177
anonymous_logon          yes
anonymous_write          yes
anonymous_login_dir      /vx/ftp_fs/anon_dir
user_logon               no
homedir_path             /vx/ftp_fs/users_dir
allow_delete             yes
security                 local
chroot_users             yes
create_homedirs          yes
```

Allows only secure (SSL) logins and rejects plain-text logins. Restart the FTP server to bring the changes into effect.

```
    FTP> set allow_non_ssl no
Changes would be applicable after restart of FTP service.
Success.

FTP> server stop
Success.

FTP> server start
Success.

FTP> show
Parameter                Current Value
---------                -------------
listen_port              21
listen_ipv6              no
max_connections          3000
```

```
max_conn_per_client      500
passive_port_range       30000:40000
idle_timeout             15 minutes
allow_non_ssl            no
umask                    177
anonymous_logon          yes
anonymous_write          yes
anonymous_login_dir      /vx/ftp_fs/anon_dir
user_logon               no
homedir_path             /vx/ftp_fs/users_dir
allow_delete             yes
security                 local
chroot_users             yes
create_homedirs          yes
```

Changes the passive port range to 35000:45000. Restart the FTP server to bring the changes into effect.

```
    FTP> set passive_port_range 35000:45000

Changes would be applicable after restart of FTP service.
Success.

FTP> server stop
Success.

FTP> server start
Success.

FTP> show
Parameter                Current Value
---------                -------------
listen_port              21
listen_ipv6              no
max_connections          3000
max_conn_per_client      500
passive_port_range       35000:45000
idle_timeout             15 minutes
allow_non_ssl            no
anonymous_logon          yes
umask                    177
anonymous_write          yes
anonymous_login_dir      /vx/ftp_fs/anon_dir
user_logon               no
homedir_path             /vx/ftp_fs/users_dir
allow_delete             yes
security                 local
chroot_users             yes
create_homedirs          yes
```

Changes the idle timeout value to 300. Restart the FTP server to bring the changes into effect.

```
    FTP> set idle_timeout 300

Changes would be applicable after restart of FTP service.
Success.

FTP> server stop
Success.
```

```
FTP> server start
Success.

FTP> show
Parameter                Current Value
---------                -------------
listen_port              21
listen_ipv6              no
max_connections          3000
max_conn_per_client      500
passive_port_range       35000:45000
idle_timeout             300 minutes
allow_non_ssl            no
umask                    177
anonymous_logon          yes
anonymous_write          yes
anonymous_login_dir      /vx/ftp_fs/anon_dir
user_logon               no
homedir_path             /vx/ftp_fs/users_dir
allow_delete             yes
security                 local
chroot_users             yes
create_homedirs          yes
```

Changes the write access to users to deny delete operations. Restart the FTP server to bring the changes into effect.

```
    FTP> set allow_delete no
Changes would be applicable after restart of FTP service.
Success.

FTP> server stop
Success.

FTP> server start
Success.

FTP> show
Parameter                Current Value
---------                -------------
listen_port              21
listen_ipv6              no
max_connections          3000
max_conn_per_client      500
passive_port_range       35000:45000
idle_timeout             300 minutes
allow_non_ssl            no
umask                    177
anonymous_logon          yes
anonymous_write          yes
anonymous_login_dir      /vx/ftp_fs/anon_dir
user_logon               no
homedir_path             /vx/ftp_fs/users_dir
allow_delete             no
security                 local
chroot_users             yes
create_homedirs          yes
```

Changes the port on which FTP service will listen for incoming connections. Restart the FTP server to bring the changes into effect.

```
    FTP> set listen_port 24
Changes would be applicable after restart of FTP service.
Success.

FTP> server stop
Success.

FTP> server start
Success.

FTP> show
Parameter                Current Value
---------                -------------
listen_port              24
listen_ipv6              no
max_connections          3000
max_conn_per_client      500
passive_port_range       35000:45000
idle_timeout             300 minutes
allow_non_ssl            no
umask                    177
anonymous_logon          yes
anonymous_write          yes
anonymous_login_dir      /vx/ftp_fs/anon_dir
user_logon               no
homedir_path             /vx/ftp_fs/users_dir
allow_delete             no
security                 local
chroot_users             yes
create_homedirs          yes
```

Changes the users' database from local to NIS/LDAP. Restart the FTP server to bring the changes into effect.

```
    FTP> set security nis_ldap
Changes would be applicable after restart of FTP service.
Success.

FTP> server stop
Success.

FTP> server start
Success.

FTP> show
Parameter                Current Value
---------                -------------
listen_port              24
listen_ipv6              no
max_connections          3000
max_conn_per_client      500
passive_port_range       35000:45000
idle_timeout             300 minutes
allow_non_ssl            no
umask                    177
anonymous_logon          yes
anonymous_write          yes
```

```
anonymous_login_dir      /vx/ftp_fs/anon_dir
user_logon               no
homedir_path             /vx/ftp_fs/users_dir
allow_delete             no
security                 nis_ldap
chroot_users             yes
create_homedirs          yes
```

Allows users to read parent directories of their home directories. Restart the FTP server to bring the changes into effect.

```
    FTP> set chroot_users no
Changes would be applicable after restart of FTP service.
Success.

FTP> server stop
Success.

FTP> server start
Success.

FTP> show
Parameter                Current Value
---------                -------------
listen_port              24
listen_ipv6              no
max_connections          3000
max_conn_per_client      500
passive_port_range       35000:45000
idle_timeout             300 minutes
allow_non_ssl            no
umask                    177
anonymous_logon          yes
anonymous_write          yes
anonymous_login_dir      /vx/ftp_fs/anon_dir
user_logon               no
homedir_path             /vx/ftp_fs/users_dir
allow_delete             no
security                 nis_ldap
chroot_users             no
create_homedirs          yes
```

Do not automatically create home directories for users when they login. Restart the FTP server to bring the changes into effect.

```
    FTP> set create_homedirs no
Changes would be applicable after restart of FTP service.
Success.

FTP> server stop
Success.

FTP> server start
Success.

FTP> show
Parameter                Current Value
---------                -------------
```

```
listen_port              24
listen_ipv6              no
max_connections          3000
max_conn_per_client      500
passive_port_range       35000:45000
idle_timeout             300 minutes
allow_non_ssl            no
umask                    177
anonymous_logon          yes
anonymous_write          yes
anonymous_login_dir      /vx/ftp_fs/anon_dir
user_logon               no
homedir_path             /vx/ftp_fs/users_dir
allow_delete             no
security                 nis_ldap
chroot_users             no
create_homedirs          no
```

8.6.5 SEE ALSO

server(1), show(1), local(1), unset(1)

8.7 show

8.7.1 SYNOPSIS

`show`

8.7.2 DESCRIPTION

The `show` command displays the list of all the configurable options and their values. These are the settings that have been configured primarily with the `set` command.

8.7.3 EXAMPLES

Display the current FTP server configuration.

```
FTP> show

Parameter                Current Value        New Value
---------                -------------        ---------
listen_port              21
listen_ipv6              no
max_connections          2000
max_conn_per_client      2000
passive_port_range       30000:40000
allow_non_ssl            yes
idle_timeout             15 minutes
umask                    177
anonymous_logon          no                   yes
anonymous_write          yes
anonymous_login_dir      /vx/fs_1
user_logon               yes
homedir_path             /vx/fs_1_test
allow_delete             yes
security                 local
chroot_users             yes
create_homedirs          yes
```

8.7.4 SEE ALSO

set(1), server(1), session(1), logupload(1)

=== unset ===

8.8 SYNOPSIS

```
unset anonymous_login_dir
unset homedir_path
```

8.9 DESCRIPTION

The `unset` commands allow the user to set the login directory of normal and anonymous users to the default value for the FTP server.

8.10 OPTIONS

`unset anonymous_login_dir` This option sets the login directory of anonymous users to its default '-'. Changes take effect when the FTP server is restarted using `server stop` followed by `server start`.

`unset homedir_path` This option sets the login directory of normal users to its default '-'. Changes take effect when the FTP server is restarted using `server stop` followed by `server start`.

8.11 EXAMPLES

Unset the login directory of the anonymous user on the FTP server. Restart the FTP server to bring the changes into effect.

```
FTP> unset anonymous_login_dir

ACCESS Ftp INFO V-288-0 Changes would be applicable after restart of FTP service.
ACCESS Ftp SUCCESS V-288-0 anonymous_login_dir set to the default '-'.


FTP> show
Parameter                Current Value        New Value
---------                -------------        ---------
listen_port              21
listen_ipv6              no
max_connections          2000
max_conn_per_client      1000
passive_port_range       30000:40000
idle_timeout             15 minutes
allow_non_ssl            yes
umask                    177
anonymous_logon          yes
anonymous_write          yes
anonymous_login_dir      /vx/ftpanon          -
user_logon               yes
homedir_path             /vx/ftphomes
allow_delete             yes
security                 local
chroot_users             yes
create_homedirs          yes

FTP> server stop
Success.

FTP> server start
Success.

FTP> show
Parameter                Current Value
---------                -------------
listen_port              21
listen_ipv6              no
max_connections          2000
max_conn_per_client      1000
passive_port_range       30000:40000
idle_timeout             15 minutes
allow_non_ssl            yes
umask                    177
anonymous_logon          yes
anonymous_write          yes
anonymous_login_dir      -
user_logon               yes
homedir_path             /vx/ftphomes
allow_delete             yes
security                 local
chroot_users             yes
create_homedirs          yes
```

Unset the login directory of the normal user (non-anonymous users) on the FTP server. Restart the FTP server to bring the changes into effect.

```
FTP> unset homedir_path

ACCESS Ftp INFO V-288-0 Changes would be applicable after restart of FTP service.
ACCESS Ftp SUCCESS V-288-0 homedir_path set to the default '-'.

FTP> server stop
Success.
FTP> server start
Success.

Parameter                Current Value
---------                -------------
listen_port              21
listen_ipv6              no
max_connections          2000
max_conn_per_client      1000
passive_port_range       30000:40000
idle_timeout             15 minutes
allow_non_ssl            yes
umask                    177
anonymous_logon          yes
anonymous_write          yes
anonymous_login_dir      -
user_logon               no
homedir_path             -
allow_delete             yes
security                 local
chroot_users             yes
create_homedirs          yes
```

8.12 SEE ALSO

server(1), show(1), local(1), set(1)

9 Network Commands

9.1 network

9.1.1 SYNOPSIS

`ping` *destination* [*nodename*] [*devicename*] [*packets*]

`dns show`

`dns enable`

`dns disable`

`firewall status`

`firewall enable`

`firewall disable`

`dns set domainname` *domainname*

`dns set nameservers` *nameserver1* [*nameserver2*] [*nameserver3*]

`dns clear domainname`

`dns clear nameservers`

`show`

`nis show` [users|groups|netgroups]

`nis disable`

`nis enable`

`nis set servername` *servername*

`nis set domainname` [*domainname*]

`nsswitch show`

`nsswitch conf` { group|hosts|netgroup|passwd|shadow }

value1 value2 value3

`ip link set` *nodename device* { up | down | mtu *MTU* }

`ip link show` [*nodename*] [*device*]

`ip addr modify` *oldipaddr newipaddr* { *netmas* | *prefix* }

`ip addr del` *ipaddr*

`ip addr add` *ipaddr* { *netmask* | *prefix* } *type* [*device*] [*nodename*]

`ip addr online` *ipaddr nodename*

`ip addr show`

`ip route show` [*nodename*]

`ip route add` *nodename ipaddr* { *netmask* | *prefix* } via *gateway* [dev *device*]

`ip route del` *nodename ipaddr* { *netmask* | *prefix* }

`ldap enable`

`ldap disable`

`ldap show` [users|groups|netgroups]

`ldap set` { server|port|basedn|binddn|ssl|rootbinddn|
users-basedn|groups-basedn|netgroups-basedn|password-hash } *value*

`ldap get` { server|port|basedn|binddn|ssl|rootbinddn|
users-basedn|groups-basedn|netgroups-basedn|password-hash }

`bond show`

`bond create` *interfacelist mode*

`bond remove` *bondname*

`netgroup add` *groupname memberlist*

`netgroup delete` *groupname* [*memberlist*]

`netgroup show` [*groupname*]

`vlan show`

`vlan add` *device vlan_id*

`vlan del` *vlan_device*

`device add` *devicename*

`device remove` *devicename*

`device rename` *old_name* `with` *new_name nodename*

`device identify` *devicename nodename* [*timeout*]

`device list` *nodename*

`device ipmonitor add` *devicename ipaddress1* [, *ipaddress2*,...]

`device ipmonitor del` *devicename*

`device ipmonitor show`

`swap` *interface1 interface2* [*nodename*]

`pciexclusion show`

`pciexclusion add` *pcilist*

`pciexclusion delete` *pci*

`krb standalone set` *kdc_server kerberos_realm keytab_file*

`krb standalone unset`

`krb standalone show`

9.1.2 DESCRIPTION

These network commands are used to configure DNS, NIS, LDAP client, IP, and check the status.

9.1.3 OPTIONS

`ping` ***destination*** **[*nodename*] [*devicename*] [*packets*]** Test whether a particular host or gateway is reachable across an IP network.

`dns show` Display the current DNS settings. Shows whether DNS is enabled or disabled, and the current values for the domain name and name servers.

`dns enable` Enable DNS usage.

`dns disable` Disable DNS usage.

`firewall status` Display whether the current firewall status is enabled or disabled.

`firewall enable` Enable the firewall setting to allow specific IPs to connect to the ports while blocking other connections.

`firewall enable` Disable the firewall setting.

`dns set domainname` ***domainname*** Set the domain name of the server to *domainname*. To clear the domain name, use `dns clear domainname`.

`dns set nameservers` ***nameserver1*** **[*nameserver2*] [*nameserver3*]** Modify the DNS name servers that are used. Specify the name servers in the order in which the name servers are to be used. Specify up to three name servers. To clear the name servers list, use the `dns clear nameservers` command.

`dns clear domainname` Clear the domain name used by DNS.

`dns clear nameservers` Clear the name servers list used by DNS.

`show` Display the network configuration and statistics on all the available nodes.

`nis show` Display the NIS server name and domain name.

`nis show users` Display NIS users that are available in the NIS database.

`nis show groups` Display NIS groups that are available in the NIS database.

`nis show netgroups` Display NIS netgroups that are available in the NIS database.

`nis disable` Disable the NIS clients on all the nodes.

`nis enable` Enable NIS clients on all the nodes.

`nis set servername` ***servername*** Set the NIS server name on all the nodes.

`nis set domainname` **[*domainname*]** Set the NIS domain name on all the nodes.

`nsswitch show` Display the name service switch configuration.

nsswitch conf { **group|hosts|netgroup|passwd|shadow** } ***value1 value2 value3*** Configure the method for the name service switch lookup process.

ip link set ***nodename device*** { **up | down | mtu** ***MTU*** } Change device attributes. Use `all` to change the attributes on all nodes.

ip link show [***nodename***] [***device***] Display device attributes.

ip addr modify ***oldipaddr newipaddr*** { ***netmask*** | ***prefix*** } Modify an IP address used by the cluster.

ip addr del ***ipaddr*** Delete an IP address.

ip addr add ***ipaddr*** { ***netmask*\|*prefix*** } ***type*** [***device***] [***nodename***] Add a new IP address.

ip addr online ***ipaddr nodename*** Bring an IP address used by the cluster online on any running node in the cluster.

ip addr show Display IP addresses and their properties.

ip route show [***nodename***] Display route entries in the routing table for the cluster.

ip route add ***nodename ipaddr*** { ***netmask*** | ***prefix*** } **via** ***gateway*** [**dev** ***device***] Add a new route.

ip route del ***nodename ipaddr*** { ***netmask*** | ***prefix*** } Delete a route used by the cluster.

ldap enable Enable the LDAP client, and configure the Pluggable Authentication Module (PAM) configuration files to use LDAP.

ldap disable Disable the LDAP client, and configure PAM configuration files not to use LDAP.

ldap show Show the LDAP client configuration.

ldap show users Show LDAP users that are available in the NSS database.

ldap show groups Show LDAP groups that are available in the NSS database.

ldap show netgroups Show LDAP netgroups that are available in the NSS database.

ldap set server ***value*** Set the LDAP server's host name or IP address. This setting is mandatory.

ldap set port ***value*** Set the LDAP server's port. Default value is 389.

ldap set basedn ***value*** Set the LDAP base Distinguished Name (DN). This setting is mandatory.

ldap set ssl { **on|off** } Set LDAP over Secure Sockets Layer (SSL) on or off. Set to "on" if your LDAP server supports SSL. Certificates required for SSL are auto-negotiated with the LDAP server during session establishment. The default value is "off." This setting is mandatory.

ldap set binddn ***value*** Set the LDAP bind DN and its password. This DN is used to bind with the LDAP server for read access. For authentication, read access to most of the attributes is required. This setting is mandatory.

ldap set rootbinddn ***value*** Set the LDAP root bind DN and its password. This DN is used to bind with the LDAP server for write access. This setting is not required for authentication. To change some attributes of an LDAP entry, this DN is required. For example, changing a user's password by root (uid=0) user requires System Administrator privileges to write into the LDAP directory. This setting is optional.

ldap set users-basedn ***value*** Set the LDAP users base DN. This DN is used by the authentication system (PAM and NSS) to search LDAP users. This setting is mandatory.

ldap set groups-basedn ***value*** Set the LDAP groups base DN. This DN is used by the authentication system (PAM and NSS) to search the LDAP groups. This setting is mandatory.

ldap set netgroups-basedn ***value*** Set the LDAP netgroups base DN. This DN is used by the authentication system (PAM and NSS) to search the LDAP netgroups. This setting is mandatory.

ldap set password-hash { **clear|crypt|md5** } Set the password hash to use when setting an LDAP user's password. The password will be encrypted with the configured hash algorithm before sending it to the LDAP server for storing into the LDAP directory. This setting is optional.

ldap get { **server|port|basedn|binddn|ssl|rootbinddn| users-basedn|groups-basedn|netgroups-basedn|password-hash** } Get the values of the configured settings.

bond show Display the bond settings.

bond create ***interfacelist mode*** { **0 | 1 | 2 | 3 | 4 | 5 | 6 |** balance-rr | active-backup | balance-xor | broadcast | 802.3ad | balance-tlb | balance-alb }:

Create the bond of listed interfaces in a specified mode.

bond remove ***bondname*** Remove the bond configuration of given bondname.

netgroup add ***groupname memberlist*** Add members to a existing netgroup or create a new netgroup.

netgroup delete ***groupname*** [***memberlist***] Delete the members of an existing netgroup or delete whole netgroup.

netgroup show [***groupname***] Display the netgroup settings.

vlan show Display the current VLAN devices.

vlan add ***device vlan_id*** Add a VLAN device with interface *device* and VLAN id as *vlan_id*.

vlan del ***vlan_device*** Delete the VLAN device *vlan_device*

device add ***devicename*** Add a NIC device *devicename* in cluster.

device remove ***devicename*** Remove NIC device *devicename* from cluster.

device rename ***old_name*** **with** ***new_name nodename*** Rename NIC device *old_name* with *new_name* on given node *nodename* of cluster.

device identify ***devicename nodename*** [***timeout***] Identify interface *devicename* by sight on given node *nodename* of cluster. Optional timeout can be specified. Default timeout is 120 sec.

device list ***nodename*** List all plugged NIC devices on node *nodename*.

device ipmonitor add ***devicename ipaddress1*** [**,** ***ipaddress2*****,...**] Add monitoring IP address(es) for public NIC devices.

device ipmonitor del ***devicename*** Delete the existing configuration of monitoring IP address(es) for a NIC device.

device ipmonitor show Display IP address(es) monitoring information for NIC devices.

swap ***interface1 interface2*** [***nodename***] Swap two network interfaces on a node in the cluster.

pciexclusion show This command displays the PCI Id those have been provided for exclusion.

pciexclusion add [***pci_list***] This commands allows administrator to add specific PCI BusId(s) for exclusion. These values must be provided before doing installation, then only it enables to exclude PCI from second node installation.

pciexclusion delete [***pci***] This commands allows administrator to delete given PCI Id from exclusion. This command must be used before doing installation then only it takes effect.

krb standalone set ***kdc_server kerberos_realm keytab_file*** This command is used to enable Kerberos authentication for NFS on Veritas Access.

krb standlaone show Display the Kerberos configuration.

krb standalone unset Reset the Kerberos configuration.

9.1.4 SEE ALSO

bond(1), dns(1), ip(1), ldap(1), netgroup(1), nis(1), nsswitch(1), ping(1), show(1), vlan(1), device(1), swap(1), pciexclusion(1), krb(1)

9.2 bond

9.2.1 SYNOPSIS

`bond show`

`bond create` *interfaceList mode*

`bond remove` *bondname*

`bond private-links-create`

`bond private-links-remove`

9.2.2 DESCRIPTION

The `bond` command is used to view, create, or remove the bonding of Ethernet interfaces.

9.2.3 OPTIONS

`interfaceList` *interfaceList* is the comma-separated list of public devices to be bonded.

`mode` Bonding of Ethernet interfaces can be done with the following types of modes

balance-rr or 0 Transmit packets in sequential order from the first available slave through the last. This mode provides load balancing and fault tolerance.:

active-backup or 1 Only one slave in the bond is active. A different slave becomes active if, and only if, the active slave fails.The bond's MAC address is externally visible on only one port (Ethernet interface) to avoid confusing the switch.

balance-xor or 2 Transmit based on the selected transmit hash policy. The default policy is a simple [(source MAC address XOR'd with destination MAC address) modulo slave count]. Alternate transmit policies may be selected via the xmit_hash_policy option. This mode provides load balancing and fault tolerance.

broadcast or 3 Transmits everything on all slave interfaces. This mode provides fault tolerance.

802.3ad or 4 Creates aggregation groups that share the same speed and duplex settings. Utilizes all slaves in the active aggregator according to the 802.3ad specification.:

balance-tlb or 5 Channel bonding that does not require any special switch support. The outgoing traffic is distributed according to the current load (computed relative to the speed) on each slave. Incoming traffic is received by the current slave. If the receiving slave fails, another slave takes over the MAC address of the failed receiving slave.

balance-alb or 6 Includes balance-tlb plus receive load balancing (RLB) for IPV4 traffic, and does not require any special switch support. The receive load balancing is achieved by ARP negotiation.:

`bond show` Show the current bonding settings.

`bond create` ***interfaceList mode*** Create the bond of devices given as interfaceList, in a given mode. During bond creation, `ssh connection may be disconnected`. The MTU of a bond interface is calculated as minimum of the MTUs of the bonded interfaces.

`bond remove` ***bondname*** Remove the bonding of Ethernet interfaces for the given bondname. During bond remove, `ssh connection may be disconnected`.

bond priv-create Create the bond for the private interfaces(priveth0 & priveth1) with mode 0 (balance-rr). `Veritas Access supports only mode 0 for the private interfaces`. During bond creation, `Veritas Access will offline all service groups, and you will need to run this command using the server console`. The MTU of a bond interface is calculated as minimum of the MTUs of the bonded interfaces.

bond priv-remove Remove the bonding of private interfaces (priveth0 & priveth1) for the cluster. During bond remove, `Veritas Access will offline all service groups, and you will need to run this command using server console`.

9.2.4 EXAMPLES

Display the bond settings.

```
    Network> bond show
BONDNAME    MODE                    DEVICES
--------    ----                    -------
bond0       active-backup   pubeth1 pubeth2

    Network> bond show
BONDNAME    MODE                    DEVICES
--------    ----                    -------
privbond0   balance-rr      priveth0 priveth1
```

Create the bond of the devices given as interfaceList in a specified mode.

```
    Network> bond create pubeth1,pubeth2 1
100% [#] Bonding interfaces. Please wait...
bond created, the bond name is: bond0
```

Remove the bond settings of the entered bondname.

```
    Network> bond remove bond0
100% [#] Removing Bond bond0. Please wait...
bond removed : bond0
```

Create the bond for private interfaces (priveth0 & priveth1). You don't need to input the mode here, as Veritas Access sets the mode by default.

```
    Network>bond priv-create
WARNING: Creating a bond on private interfaces will OFFLINE all services. Verify all␣
↪private interfaces are connected using a switch.
Do you want to continue (y/n) y
100% [#] Bonding interfaces. Please wait...
Private interfaces bond created, the bondname is: privbond0.
```

Remove the bond settings for the private interface bond.

```
    Network> bond priv-remove
WARNING: Removing a bond on private interfaces will OFFLINE all services.
Do you want to continue (y/n) y
100% [#] Removing private interface bond privbond0. Please wait...
Private interfaces bond removed : privbond0
```

Change in MTU during bond creation.

```
    Network> ip link show
Nodename                Device  Status  MTU  Detect  Speed      HWaddr
--------                ------  ------  ---  ------  ------     ------
node5_1                pubeth0    UP    1500  yes    100Mb/s    00:0c:29:a8:9d:f3
node5_1                pubeth1    UP    1000  yes    100Mb/s    00:0c:29:a8:9d:fd
node5_2                pubeth0    UP    1600  yes    100Mb/s    00:0c:29:da:c9:e2
node5_2                pubeth1    UP    1600  yes    100Mb/s    00:0c:29:da:c9:ec
    Network> bond create pubeth0,pubeth1 1
100% [#] Bonding interfaces. Please wait...
bond created, the bond name is: bond0
    Network> ip link show
Nodename                Device  Status  MTU  Detect  Speed      HWaddr
--------                ------  ------  ---  ------  ------     ------
node5_1                pubeth0    UP    1000  yes    100Mb/s    00:0c:29:a8:9d:f3
node5_1                pubeth1    UP    1000  yes    100Mb/s    00:0c:29:a8:9d:fd
node5_1                bond0      UP    1000  -      -          00:0c:29:a8:9d:f3
node5_2                pubeth0    UP    1600  yes    100Mb/s    00:0c:29:da:c9:e2
node5_2                pubeth1    UP    1600  yes    100Mb/s    00:0c:29:da:c9:ec
node5_2                bond0      UP    1600  -      -          00:0c:29:da:c9:e2
```

9.2.5 SEE ALSO

dns(1), ip(1), ldap(1), netgroup(1), nis(1), nsswitch(1), ping(1), show(1), vlan(1), device(1), swap(1), pciexclusion(1)

9.3 device

9.3.1 SYNOPSIS

`device add` *devicename*

`device remove` *devicename*

`device rename` *old_name* `with` *new_name nodename*

`device identify` *devicename nodename* [*timeout*]

`device list` *nodename*

`device ipmonitor add` *devicename ipaddress1*[,*ipaddress2,...*]

`device ipmonitor del` *devicename*

`device ipmonitor show`

9.3.2 DESCRIPTION

The network `device` command is used to add, remove, rename, identify and list NIC devices from Veritas Accesscluster.

A new NIC device or an existing unused/excluded device can be added to Veritas Accesscluster by using the `device add` command. The given device must be present on all the nodes of the cluster.

`device remove` command removes the device from cluster control, and if required it can be unplugged from cluster node. When device is removed, all physical and virtual IPs will be deleted from that NIC device. All physical IPs will be kept in free list and will be available for reuse, virtual IPs will not be available for reuse. One need to re-add virtual IPs in case of reuse. After deleting device from Veritas Accesscluster it will be renamed with eth#, where # is lowest available number for interface index in name 'eth'. And because of this new name may not be same as old eth# name.

Device names can be changed by using the `device rename` command. Only device of name prefix 'eth' can be renamed. Device with new name should not be present on any node of the cluster. In case of mismatches in names of newly added NICs in cluster, one can rename those devices and then add devices in cluster.

`device identify` command will start visual indicators on NIC hardware if supported and available. This helps to locate the actual, physical NIC on the node. By default the timeout value is 120 seconds.

`device list` command will print bus IDs and MAC addresses of all the devices on the given node irrespective of NICs PCI exclusion state.

`device ipmonitor add` command will add monitoring IP address(es) for public NIC devices. Use `all` to configure *ipmonitor* for all public devices. A device is considered to be in working condition if it is able to ping at lease one of the monitoring IP addresses.

`device ipmonitor del` command will delete the existing configuration of monitoring IP address(es) for a NIC device. Use `all` to delete *ipmonitor* for all devices.

`device ipmonitor show` command will display IP address(es) monitoring information for NIC devices along with the state of the NIC devices.

9.3.3 OPTIONS

nodename The node on which the operation should takes place.

devicename Device on which the operation should takes place.

old_name Present interface name.

new_name Expected new interface name.

timeout Input time in seconds.

9.3.4 EXAMPLES

Display the devices.

```
    Network> device list cls_01
Device      BusID            MAC Addr              Device Info
------      -----            --------              -----------
priveth0    0000:02:00.0     00:50:56:be:00:9e     82545EM Gigabit Ethernet Controller␣
↪(Copper)
priveth1    0000:02:01.0     00:50:56:be:00:9f     82545EM Gigabit Ethernet Controller␣
↪(Copper)
pubeth0     0000:02:02.0     00:50:56:be:00:a0     82545EM Gigabit Ethernet Controller␣
↪(Copper)
pubeth1     0000:02:03.0     00:50:56:be:00:a1     82545EM Gigabit Ethernet Controller␣
↪(Copper)
eth1        0000:02:05.0     00:0c:29:7d:d2:8a     82545EM Gigabit Ethernet Controller␣
↪(Copper)
eth2        0000:02:06.0     00:0c:29:7d:d2:94     82545EM Gigabit Ethernet Controller␣
↪(Copper)
```

To add a device with the name eth2.

```
    Network> device add eth2

Network> device list cls_01
Device      BusID            MAC Addr              Device Info
------      -----            --------              -----------
priveth0    0000:02:00.0     00:50:56:be:00:9e     82545EM Gigabit Ethernet Controller␣
↪(Copper)
priveth1    0000:02:01.0     00:50:56:be:00:9f     82545EM Gigabit Ethernet Controller␣
↪(Copper)
pubeth0     0000:02:02.0     00:50:56:be:00:a0     82545EM Gigabit Ethernet Controller␣
↪(Copper)
pubeth1     0000:02:03.0     00:50:56:be:00:a1     82545EM Gigabit Ethernet Controller␣
↪(Copper)
eth1        0000:02:05.0     00:0c:29:7d:d2:8a     82545EM Gigabit Ethernet Controller␣
↪(Copper)
pubeth2     0000:02:06.0     00:0c:29:7d:d2:94     82545EM Gigabit Ethernet Controller␣
↪(Copper)
```

To delete a device pubeth2 that is shown in the output of *device list*.

```
    Network> device remove pubeth2

Network> device list cls_01
Device      BusID            MAC Addr              Device Info
------      -----            --------              -----------
priveth0    0000:02:00.0     00:50:56:be:00:9e     82545EM Gigabit Ethernet Controller␣
↪(Copper)
priveth1    0000:02:01.0     00:50:56:be:00:9f     82545EM Gigabit Ethernet Controller␣
↪(Copper)
pubeth0     0000:02:02.0     00:50:56:be:00:a0     82545EM Gigabit Ethernet Controller␣
↪(Copper)
```

```
pubeth1     0000:02:03.0    00:50:56:be:00:a1    82545EM Gigabit Ethernet Controller␣
↪(Copper)
eth1        0000:02:05.0    00:0c:29:7d:d2:8a    82545EM Gigabit Ethernet Controller␣
↪(Copper)
eth2        0000:02:06.0    00:0c:29:7d:d2:94    82545EM Gigabit Ethernet Controller␣
↪(Copper)
```

To identify a device eth1 on cluster node 1 with a timeout of 60 seconds.

```
Network> device identify eth1 cls_01 60
```

To rename device eth1 with eth0 on cluster node 1.

```
Network> device rename eth1 with eth0 cls_01
```

To add a monitoring IP addresses for device pubeth0.

```
Network> device ipmonitor add pubeth0 10.29.29.29,10.29.29.30
```

To delete the monitoring IP addresses for all public devices.

```
Network> device ipmonitor del all
```

To show the monitoring IP addresses for all public devices.

```
    Network> device ipmonitor show
Device          Monitor IPs                    dsga_01        dsga_02
------          -----------                    --------       --------
pubeth0         10.209.107.55                  ONLINE         ONLINE
pubeth1         10.209.107.55                  ONLINE         ONLINE
```

9.3.5 SEE ALSO

bond(1), dns(1), ip(1), ldap(1), netgroup(1), nis(1), nsswitch(1), ping(1), show(1), vlan(1), swap(1), pciexclusion(1)

9.4 dns

9.4.1 SYNOPSIS

`dns show`

`dns enable`

`dns disable`

`dns set domainname` *domainname*

`dns set nameservers` *nameserver1* [*nameserver2*] [*nameserver3*]

`dns set searchdomains` *searchdomain1* [,*searchdomain2*] [,*searchdomain3*]

`dns clear domainname`

`dns clear nameservers`

9.4.2 DESCRIPTION

The network `dns` command is used to view or change the DNS settings.

9.4.3 OPTIONS

`dns show` Show the current DNS settings. Show whether DNS is enabled or disabled, and the current values for the domain name and name servers.

`dns enable` Enable DNS usage.

`dns disable` Disable DNS usage.

`dns set domainname` *domainname* Set the domain name of the server to *domainname*. To clear the domain name, use `dns clear domainname`.

`dns set nameservers` *nameserver1* [*nameserver2*] [*nameserver3*] Modify the DNS name servers that are used. Specify the name servers in the order in which the name servers should be used. Specify up to three name servers. To clear the name servers list, use `dns clear nameservers`.

`dns set searchdomains` *searchdomain1*[,searchdomain2*][,*searchdomain3*] Modify the DNS search domains that are used. Specify the search domains in the order in which the search domains should be used.

`dns clear domainname` Clear the domain name used by DNS.

`dns clear nameservers` Clear the name servers list used by DNS.

9.4.4 EXAMPLES

Display the current DNS settings.

```
    Network> dns show
DNS Status  : Enabled
domain              : cluster1.com
nameserver  : 10.216.50.132
```

Show and then enable the usage of DNS.

```
    Network> dns show
DNS Status      : Disabled
Old Settings
------------
domain              : cluster1.com
nameserver  : 10.216.50.132
    network> dns enable
    network> dns show
DNS Status  : Enabled
domain              : cluster1.com
nameserver  : 10.216.50.132
```

Disable the usage of DNS.

```
    Network> dns disable
    Network> dns show
DNS Status  : Disabled
Old Settings
------------
domain              : cluster1.com
nameserver  : 10.216.50.132
```

Change the DNS domainname.

```
    Network> dns set domainname example.com
    Network> dns show
DNS Status      : Enabled
domain          : example.com
nameserver      : 10.216.50.132
```

Clear the domain name being used.

```
    Network> dns clear domainname
    Network> dns show
DNS Status      : Enabled
nameserver      : 10.216.50.132
```

Set the name servers as 10.216.50.199 and 10.216.50.200.

```
    Network> dns set nameservers 10.216.50.199 10.216.50.200
    Network> dns show
DNS Status  : Enabled
nameserver  : 10.216.50.199
nameserver  : 10.216.50.200
```

Remove all name servers that are being used.

```
    Network> dns clear nameservers
    Network> dns show
DNS Status      : Enabled
```

9.4.5 SEE ALSO

bond(1), ip(1), ldap(1), netgroup(1), nis(1), nsswitch(1), ping(1), show(1), vlan(1), device(1), swap(1), pciexclusion(1)

9.5 firewall

9.5.1 SYNOPSIS

`firewall status`

`firewall enable`

`firewall disable`

9.5.2 DESCRIPTION

The network `firewall` command is used to view or change the firewall settings.

9.5.3 OPTIONS

`firewall status` Display whether the current firewall status is enabled or disabled.

`firewall enable` Enable the firewall setting to allow specific IPs to connect to the ports while blocking other connections.

`firewall disable` Disable the firewall setting and allow connections on any port from any IP. Applied rules do not work when the firewall setting is disabled.

9.5.4 EXAMPLES

Display the current firewall settings.

```
    Network> firewall status
Firewall status : DISABLED
```

Enable firewall.

```
    Network> firewall enable
ACCESS net INFO V-288-0 Firewall successfully enabled
```

Disable firewall.

```
    Network> firewall disable
It is not advisable to disable firewall. Do you want you continue (y/n): y
ACCESS net INFO V-288-0 Firewall successfully disabled
```

9.5.5 SEE ALSO

bond(1), ip(1), nis(1), nsswitch(1), ping(1), show(1), device(1)

9.6 ip

9.6.1 SYNOPSIS

`ip link set` *nodename device* { up | down | mtu *MTU* }

`ip link show` [*nodename*] [*device*]

`ip addr modify` *oldipaddr newipaddr* { *netmask* | *prefix* }

`ip addr del` *ipaddr*

`ip addr add` *ipaddr* { *netmask* | *prefix* } *type* [*device*] [*nodename*]

`ip addr online` *ipaddr nodename*

`ip addr show`

`ip route show` [*nodename*]

`ip route add` *nodename ipaddr* { *netmask* | *prefix* }

via *gateway* [dev *device*]

`ip route del` *nodename ipaddr* { *netmask* | *prefix* }

9.6.2 OPTIONS

nodename Node on which the operation takes place.g A value of `all` indicates the operation takes place on all nodes of the cluster.

device Device on which the operation takes place.

ipaddr Specifies the IP address. You can specify either an IPv4 address, or an IPv6 address.

oldipaddr Specifies the old IP address to be modified.g

newipaddr Specifies what the new IP address will be.

netmask Netmask for the IPv4 address. Specify an IPv4 address in the format AAA.BBB.CCC.DDD, where each number ranges from 0 to 255.

prefix Prefix length for the IPv6 address. The accepted range is 0-128 integers.

`up` Changes the state of the device to `up`.

`down` Changes the state of the device to `down`.

`mtu` *MTU* The Maximum Transmission Unit (MTU) of the device.

type Specifies how the *ipaddr* should be used by the cluster. Theg *type* can be `physical` or `virtual`. Physical IP addresses are bound to an interface. Virtual IP addresses can be moved from one interface to other. NFS is served using the virtual IP addresses.

`ip link set` *nodename device* { up | down | mtu *MTU* } Change network device attributes or states. Use `all` to change attributes on all nodes. Devices that are part of a bond interface can not be configured through this command. MTU changes to a bond interface get applied to all the interfaces that are part of that bond. Note: Setting the incorrect MTU value will cause the console ip to be unavailable.

`ip link show` [*nodename*] [*device*] Display device attributes.

ip addr add ***ipaddr netmask|prefix type*** **[*device*] [*nodename*]** Add a new IP address. IP is a protocol that allows addresses to be attached to a network device. Each device must have at least one address to use the corresponding protocol. Both IPv4 and IPv6g addresses can be attached to one device. The *type* can be `physical` or `virtual` The *device* attribute can be used with IP *type* `virtual` If the value *type* is `physical`, and the value of *device* and *nodename* are valid given physical IP will be added for input device. If no *device* and *nodename* configured input physical IP will be added to free list.

ip addr del ***ipaddr*** Delete an IP address from the cluster. Physical IP addresses can be deleted only if they are not being used by any interface of any node in the cluster, or either type of physical IP is present on interface. Minimum one type of physical IP should be present on interface. Virtual IP addresses, except the console IP address, can be deleted.

ip addr modify ***oldipaddr newipaddr netmask|prefix*** Modify an IP address used by the cluster. Physical IP addresses and virtual IP addresses can both be modified.g Virtual IP address modifications cause NFS connections on *oldipaddr* to be terminated.

ip addr online ***ipaddr nodename*** Bring an IP address online for a running node of the cluster.

ip addr show Shows the IP addresses being used.

ip route add ***nodename ipaddr netmask|prefix*** **via** ***gateway*** **[dev** ***device*****]** Add a new route for the cluster. The routing table contains information about paths to other networked nodes. Routing table changes can be made per node of the cluster. Use `all` for *nodename* to add the route to all nodes of the cluster. Use a netmask(prefix for IPv6) value of `255.255.255.255(128 for IPv6)` for the *netmask* to add a host route to *ipaddr*. Use a value of `0.0.0.0(:: for IPv6)` for the *gateway* to add a route that does not use any gateway. The `dev` *device* is an optional argument.g Use any of the public devices for the *device* (pubeth0, pubeth1, or any).

ip route del ***nodename ipaddr netmask|prefix*** Delete a route used by the cluster. Use `all` for *nodename* to delete the route from all nodes of the cluster. The combination of *ipaddr* and *netmask* specifies the network or hostg for which the route is deleted.g Use a value of `255.255.255.255(128 for IPv6)` for the *netmask|prefix* to delete a host route to *ipaddr*.

ip route show [*nodename*] Display the routing table of the nodes in the cluster. Use `all` to see the routing table from all nodes of the cluster.

9.6.3 EXAMPLES

Display the current link attributes and states.

```
Network> ip link show
Nodename                Device  Status  MTU  Detect  Speed      HWaddr
--------                ------  ------  ---  ------  -----      ------
node5_1                pubeth0    UP    1500  yes    100Mb/s    00:0c:29:a8:9d:f3
node5_1                pubeth1    UP    1500  yes    100Mb/s    00:0c:29:a8:9d:fd
node5_2                pubeth0    UP    1500  yes    100Mb/s    00:0c:29:da:c9:e2
node5_2                pubeth1    UP    1500  yes    100Mb/s    00:0c:29:da:c9:ec
```

View the attributes and state of a specific node and a specific interface.

```
Network> ip link show node5_2 pubeth0
Nodename                Device  Status  MTU  Detect  Speed      HWaddr
--------                ------  ------  ---  ------  -----      ------
node5_2                pubeth0    UP    1500  yes    100Mb/s    00:0c:29:a8:9d:f3
```

Set the MTU value on all nodes for a specific device.

```
Network> ip link set all pubeth0 mtu 1600
Network> ip link show
Nodename                Device  Status  MTU  Detect  Speed      HWaddr
--------                ------  ------  ---  ------  -----      ------
node5_1                pubeth0    UP    1600  yes    100Mb/s    00:0c:29:a8:9d:f3
node5_1                pubeth1    UP    1500  yes    100Mb/s    00:0c:29:a8:9d:fd
node5_2                pubeth0    UP    1600  yes    100Mb/s    00:0c:29:da:c9:e2
node5_2                pubeth1    UP    1500  yes    100Mb/s    00:0c:29:da:c9:ec
```

Display the IP addresses used by the cluster and their states.

```
Network> ip addr show
IP              Netmask         Device     Node        Type     Status
--              -------         ------     ----        ----     ------
10.216.114.212  255.255.248.0   pubeth0    node5_1     Physical
10.216.114.213  255.255.248.0   pubeth1    node5_1     Physical
10.216.114.214  255.255.248.0   pubeth0    node5_2     Physical
10.216.114.215  255.255.248.0   pubeth1    node5_2     Physical
10.216.114.217  255.255.248.0   pubeth0    node5_1     Virtual  ONLINE (Con IP)
10.10.10.10     255.255.248.0   pubeth0    node5_1     Virtual  ONLINE
10.10.10.11     255.255.248.0   pubeth1    node5_1     Virtual  ONLINE
10.10.10.12     255.255.248.0   pubeth0    node5_2     Virtual  ONLINE
10.10.10.13     255.255.248.0   pubeth1    node5_2     Virtual  ONLINE
```

Add an IP address to the physcial IP addresses used by the cluster.

```
Network> ip addr add 10.216.114.216 255.255.248.0 physical
Network> ip addr show
IP              Netmask         Device    Node         Type     Status
--              -------         ------    ----         ----     ------
10.216.114.212  255.255.248.0   pubeth0   node5_1      Physical
10.216.114.213  255.255.248.0   pubeth1   node5_1      Physical
10.216.114.214  255.255.248.0   pubeth0   node5_2      Physical
10.216.114.215  255.255.248.0   pubeth1   node5_2      Physical
10.216.114.216  255.255.248.0           ( unused )     Physical
10.216.114.217  255.255.248.0   pubeth0   node5_1      Virtual  ONLINE (Con IP)
10.10.10.10     255.255.248.0   pubeth0   node5_1      Virtual  ONLINE
10.10.10.11     255.255.248.0   pubeth1   node5_1      Virtual  ONLINE
10.10.10.12     255.255.248.0   pubeth0   node5_2      Virtual  ONLINE
10.10.10.13     255.255.248.0   pubeth1   node5_2      Virtual  ONLINE
```

Add an IP address to the virtual IP addresses used by the cluster.

```
Network> ip addr add 10.10.10.14 255.255.248.0 virtual
Network> ip addr show
IP              Netmask         Device    Node         Type     Status
--              -------         ------    ----         ----     ------
10.216.114.212  255.255.248.0   pubeth0   node5_1      Physical
10.216.114.213  255.255.248.0   pubeth1   node5_1      Physical
10.216.114.214  255.255.248.0   pubeth0   node5_2      Physical
10.216.114.215  255.255.248.0   pubeth1   node5_2      Physical
10.216.114.216  255.255.248.0           ( unused )     Physical
10.216.114.217  255.255.248.0   pubeth0   node5_1      Virtual  ONLINE (Con IP)
10.10.10.10     255.255.248.0   pubeth0   node5_1      Virtual  ONLINE
10.10.10.11     255.255.248.0   pubeth1   node5_1      Virtual  ONLINE
10.10.10.12     255.255.248.0   pubeth0   node5_2      Virtual  ONLINE
10.10.10.13     255.255.248.0   pubeth1   node5_2      Virtual  ONLINE
10.10.10.14     255.255.248.0   pubeth0   node5_1      Virtual  ONLINE
```

Add an IP address to the virtual IP addresses used by the cluster for a VLAN device

```
Network> ip addr add 10.10.10.14 255.255.248.0 virtual pubeth1.2
Network> ip addr show
IP              Netmask         Device    Node         Type     Status
--              -------         ------    ----         ----     ------
10.216.114.212  255.255.248.0   pubeth0   node5_1      Physical
10.216.114.213  255.255.248.0   pubeth1   node5_1      Physical
10.216.114.214  255.255.248.0   pubeth0   node5_2      Physical
10.216.114.215  255.255.248.0   pubeth1   node5_2      Physical
10.216.114.216  255.255.248.0   pubeth1.2 node5_1      Physical
10.216.114.217  255.255.248.0   pubeth0.2 node5_2      Physical
10.216.114.218  255.255.248.0           ( unused )     Physical
10.216.114.219  255.255.248.0   pubeth0   node5_1      Virtual  ONLINE (Con IP)
10.10.10.10     255.255.248.0   pubeth0   node5_1      Virtual  ONLINE
10.10.10.11     255.255.248.0   pubeth1   node5_1      Virtual  ONLINE
10.10.10.12     255.255.248.0   pubeth0   node5_2      Virtual  ONLINE
10.10.10.13     255.255.248.0   pubeth1   node5_2      Virtual  ONLINE
10.10.10.14     255.255.248.0   pubeth1.2 node5_1      Virtual  ONLINE
```

Delete an IP address from the physical IP addresses being used by the cluster. Physical IP addresses that are assigned to a device cannot be deleted; they can only be modified.g The virtual IP address for the console cannot be deleted; it can only be modified.

```
Network> ip addr del 10.216.114.216g
Network> ip addr show
IP              Netmask         Device     Node       Type     Status
--              -------         ------     ----       ----     ------
10.216.114.212  255.255.248.0   pubeth0   node5_1     Physical
10.216.114.213  255.255.248.0   pubeth1   node5_1     Physical
10.216.114.214  255.255.248.0   pubeth0   node5_2     Physical
10.216.114.215  255.255.248.0   pubeth1   node5_2     Physical
10.216.114.217  255.255.248.0   pubeth0   node5_1     Virtual  ONLINE (Con IP)
10.10.10.10     255.255.248.0   pubeth0   node5_1     Virtual  ONLINE
10.10.10.11     255.255.248.0   pubeth1   node5_1     Virtual  ONLINE
10.10.10.12     255.255.248.0   pubeth0   node5_2     Virtual  ONLINE
10.10.10.13     255.255.248.0   pubeth1   node5_2     Virtual  ONLINE
10.10.10.14     255.255.248.0   pubeth0   node5_1     Virtual  ONLINE
```

Modify an IP address from the IP addresses being used by the cluster. Physical IP addresses which are assigned to a device cannot be deleted; they can only be modified.g The virtual IP address for the console cannot be deleted; it can only be modified.

```
Network> ip addr modify 10.10.10.14 10.10.10.15 255.255.248.0
Network> ip addr show
IP              Netmask         Device    Node        Type     Status
--              -------         ------    ----        ----     ------
10.216.114.212  255.255.248.0   pubeth0   node5_1     Physical
10.216.114.213  255.255.248.0   pubeth1   node5_1     Physical
10.216.114.214  255.255.248.0   pubeth0   node5_2     Physical
10.216.114.215  255.255.248.0   pubeth1   node5_2     Physical
10.216.114.217  255.255.248.0   pubeth0   node5_1     Virtual  ONLINE (Con IP)
10.10.10.10     255.255.248.0   pubeth0   node5_1     Virtual  ONLINE
10.10.10.11     255.255.248.0   pubeth1   node5_1     Virtual  ONLINE
10.10.10.12     255.255.248.0   pubeth0   node5_2     Virtual  ONLINE
10.10.10.13     255.255.248.0   pubeth1   node5_2     Virtual  ONLINE
10.10.10.15     255.255.248.0   pubeth0   node5_1     Virtual  ONLINE
```

Online an IP protocol address used by the cluster on any running node of the cluster.

```
Network> ip addr online 10.10.10.15  node5_2
Network> ip addr show
IP              Netmask         Device    Node        Type     Status
--              -------         ------    ----        ----     ------
10.216.114.212  255.255.248.0   pubeth0   node5_1     Physical
10.216.114.213  255.255.248.0   pubeth1   node5_1     Physical
10.216.114.214  255.255.248.0   pubeth0   node5_2     Physical
10.216.114.215  255.255.248.0   pubeth1   node5_2     Physical
10.216.114.217  255.255.248.0   pubeth0   node5_1     Virtual  ONLINE (Con IP)
10.10.10.10     255.255.248.0   pubeth0   node5_1     Virtual  ONLINE
10.10.10.11     255.255.248.0   pubeth1   node5_1     Virtual  ONLINE
10.10.10.12     255.255.248.0   pubeth0   node5_2     Virtual  ONLINE
10.10.10.13     255.255.248.0   pubeth1   node5_2     Virtual  ONLINE
10.10.10.15     255.255.248.0   pubeth0   node5_2     Virtual  ONLINE
```

View the routing table of a specific node in the cluster.

```
Network> ip route show node5_2
node5_2
----------------
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
172.16.89.0     0.0.0.0         255.255.255.0   U         0 0          0 priveth0
10.216.112.0    0.0.0.0         255.255.248.0   U         0 0          0 pubeth0
10.216.112.0    0.0.0.0         255.255.248.0   U         0 0          0 pubeth1
10.254.0.0      0.0.0.0         255.255.0.0     U         0 0          0 pubeth0
127.0.0.0       0.0.0.0         255.0.0.0       U         0 0          0 lo
0.0.0.0         10.216.112.1    0.0.0.0         UG        0 0          0 pubeth0
```

Add a route through a given device to the routing table of all nodes in the cluster. Use a value of 0.0.0.0 for the gateway, to use a route that is based on only the device and does not use any gateway.

```
Network> ip route add all 10.216.128.0 255.255.248.0 via 0.0.0.0
Network> ip route show
node5_1
----------------
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
172.16.89.0     0.0.0.0         255.255.255.0   U         0 0          0 priveth0
10.216.112.0    0.0.0.0         255.255.248.0   U         0 0          0 pubeth0
10.216.112.0    0.0.0.0         255.255.248.0   U         0 0          0 pubeth1
10.254.0.0      0.0.0.0         255.255.0.0     U         0 0          0 pubeth0
127.0.0.0       0.0.0.0         255.0.0.0       U         0 0          0 lo
0.0.0.0         10.216.112.1    0.0.0.0         UG        0 0          0 pubeth0

node5_2
----------------
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
172.16.89.0     0.0.0.0         255.255.255.0   U         0 0          0 priveth0
10.216.112.0    0.0.0.0         255.255.248.0   U         0 0          0 pubeth0
10.216.112.0    0.0.0.0         255.255.248.0   U         0 0          0 pubeth1
10.254.0.0      0.0.0.0         255.255.0.0     U         0 0          0 pubeth0
127.0.0.0       0.0.0.0         255.0.0.0       U         0 0          0 lo
0.0.0.0         10.216.112.1    0.0.0.0         UG        0 0          0 pubeth0
```

Delete the default gateway being used by a specific node. The deletion only aim at static routes which are added by command (`ip route add`).

```
Network> ip route del node5_2  10.216.128.0 255.255.248.0 via 0.0.0.0
Network> ip route show node5_2
node5_2
----------------
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
172.16.89.0     0.0.0.0         255.255.255.0   U         0 0          0 priveth0
10.216.112.0    0.0.0.0         255.255.248.0   U         0 0          0 pubeth0
10.216.112.0    0.0.0.0         255.255.248.0   U         0 0          0 pubeth1
10.254.0.0      0.0.0.0         255.255.0.0     U         0 0          0 pubeth0
127.0.0.0       0.0.0.0         255.0.0.0       U         0 0          0 lo
```

9.6.4 SEE ALSO

bond(1), dns(1), ldap(1), netgroup(1), nis(1), nsswitch(1), ping(1), show(1), vlan(1), device(1), swap(1), pciexclusion(1)

9.7 krb

9.7.1 SYNOPSIS

`krb standalone set` *kdc_server kerberos_realm keytab_file*

`krb standalone unset`

`krb standalone show`

9.7.2 OPTIONS

kdc_server Kerberos KDC server.

kerberos_realm Kerberos realm.

keytab_file Location of keytab file on the console node.

`krb standalone set` ***kdc_server kerberos_realm keytab_file*** This command enables Kerberos authentication for NFS mounts. This would set Kerberos configuration on Veritas Access given the KDC server, Kerberos realm and the keytab file. Keytab file should have been uploaded to the Veritas Access console node. Keytab file will be copied from the console node to /etc/krb5.keytab on all of the nodes of the cluster. Kerberos will be configured on all of the nodes of the cluster. Keytab file should have been set up with clustername as the hostname. Kerberos cannot be configured with standalone KDC if CIFS is already configured with `ads`. This command also checks for the correct domain in the **/etc/idmapd.conf** file. If the domain is not set, NFSv4 idmapping will not be proper and the command will throw a warning to set the domain.

`krb standlaone show` Display the Kerberos configuration.

`krb standalone unset` Reset the Kerberos configuration.

9.7.3 EXAMPLES

```
test_fa.Network> krb standalone set kdc_server TESTKDC.COM /home/support/krb5.keytab
ACCESS krb SUCCESS V-288-999 Kerberos configured for NFS

test_fa.Network> krb standalone show
Kerberos General Info:
======================
KDC:    kdc_server
REALM:  TESTKDC.COM

Keytab Info:
============
Keytab name: FILE:/etc/krb5.keytab
KVNO Timestamp         Principal
---- ----------------- --------------------------------------------------------
   2 07/07/15 16:02:54 nfs/test_fa@TESTKDC.COM (aes256-cts-hmac-sha1-96)
   2 07/07/15 16:02:54 nfs/test_fa@TESTKDC.COM (aes128-cts-hmac-sha1-96)
   2 07/07/15 16:02:54 nfs/test_fa@TESTKDC.COM (des3-cbc-sha1)
   2 07/07/15 16:02:54 nfs/test_fa@TESTKDC.COM (arcfour-hmac)
   2 07/07/15 16:02:54 nfs/test_fa@TESTKDC.COM (des-hmac-sha1)
   2 07/07/15 16:02:55 nfs/test_fa@TESTKDC.COM (des-cbc-md5)

test_fa.Network> krb standalone unset
```

```
ACCESS krb SUCCESS V-288-999 Kerberos configuration is reset

test_fa.Network> krb standalone show
Kerberos is not configured
test_fa.Network>
```

9.8 ldap

9.8.1 SYNOPSIS

`ldap enable`

`ldap disable`

`ldap show` [users|groups|netgroups]

`ldap set` { server|port|basedn|binddn|ssl|rootbinddn|

users-basedn|groups-basedn|netgroups-basedn|password-hash } *value*

`ldap get` { server|port|basedn|binddn|ssl|rootbinddn|

users-basedn|groups-basedn|netgroups-basedn|password-hash }

`ldap clear` { server|port|basedn|binddn|ssl|rootbinddn|

users-basedn|groups-basedn|netgroups-basedn|password-hash }

9.8.2 DESCRIPTION

The network `ldap` commands are used to configure the Lightweight Directory Access Protocol (LDAP) client for authentication. These commands configure the system to use LDAP users and groups when logging into the system. LDAP clients use the LDAPv3 protocol for communicating with the server. Enabling the LDAP client configures the Pluggable Authentication Module (PAM) for LDAP users and groups. PAM is the standard authentication framework for Linux.

9.8.3 OPTIONS

value Value of the variable to set.

`ldap enable` Enable the LDAP client, and configure PAM configuration files to use LDAP.

`ldap disable` Disable the LDAP client, and configure PAM configuration files not to use LDAP.

`ldap show` Display the LDAP client configuration.

`ldap show users` Display the LDAP users that are available in the NSS database.

`ldap show groups` Display the LDAP groups that are available in the NSS database.

`ldap show netgroups` Display the LDAP netgroups that are available in the NSS database.

`ldap set server` ***value*** Set the LDAP server's host name or IP address. This setting is mandatory.

`ldap set port` ***value*** Set the LDAP server's port. Default value is 389.

`ldap set basedn` ***value*** Set the LDAP base Distinguished Name (DN). This setting is mandatory.

`ldap set ssl` **{ on|off }** Set LDAP over Secure Sockets Layer (SSL) to be on or off. Set this option to "on" if your LDAP server supports SSL. Certificates required for SSL are auto-negotiated with the LDAP server during session establishment. The default value is "off." This setting is mandatory.

`ldap set binddn` ***value*** Set the LDAP bind DN and its password. This DN is used to bind with the LDAP server for "read" access. For authentication, "read" access to most of the attributes is required. This setting is mandatory.

ldap set rootbinddn *value* Set the LDAP root bind DN and its password. This DN is used to bind with the LDAP server for "write" access. This setting is not required for authentication. To change some attributes of an LDAP entry, this DN is required. For example, changing a user's password by root (uid=0) user requires admin privileges to write into the LDAP directory. This setting is optional.

ldap set users-basedn *value* Set the LDAP users base DN. This DN is used by the authentication system (PAM and NSS) to search LDAP users. This setting is mandatory.

ldap set groups-basedn *value* Set the LDAP groups base DN. This DN is used by the authentication system (PAM and NSS) to search the LDAP groups. This setting is mandatory.

ldap set netgroups-basedn *value* Set the LDAP netgroups base DN. This DN is used by the authentication system (PAM and NSS) to search the LDAP netgroups. This setting is mandatory.

ldap set password-hash { clear|crypt|md5 } Set the password hash to use when setting an LDAP user's password. The password will be encrypted with the configured hash algorithm before sending it to the LDAP server for storing into the LDAP directory. This setting is optional.

ldap get { server|port|basedn|binddn|ssl|rootbinddn|users-basedn|groups-basedn|netgroups-basedn|password-hash } Get the values of the configured settings.

ldap clear { server|port|basedn|binddn|ssl|rootbinddn| users-basedn|groups-basedn|netgroups-basedn|password-hash }:

Clears a specified attribute and sets to default.

9.8.4 EXAMPLES

Enable the LDAP client.

```
Network> ldap enable
```

Disable the LDAP client.

```
Network> ldap disable
```

Display the LDAP client configuration.

```
    Network> ldap show
LDAP client is enabled.
=======================
LDAP server:               ldap_server
LDAP port:                 389 (default)
LDAP base DN:              dc=example,dc=com
LDAP over SSL:             on
LDAP bind DN:              cn=binduser,dc=example,dc=com
LDAP root bind DN:         cn=admin,dc=example,dc=com
LDAP password hash:        md5
LDAP users base DN:        ou=Users,dc=example,dc=com
LDAP groups base DN:       ou=Groups,dc=example,dc=com
LDAP netgroups base DN:    ou=Netgroups,dc=example,dc=com
OK Completed
```

Display the LDAP users available to the authentication system.

```
    Network> ldap show users
=================
User 'ldapuser1':
=================
```

```
Username:       Username Surname
uid:            2000
gid:            2000
Home Directory: /home/ldapuser1
Login Shell:    /bin/bash
```

Display the LDAP groups available to the authentication system.

```
    Network> ldap show groups
=============
Group '2000':
=============
Groupname:              group1
Members of the Group:  ldapuser1
```

Show LDAP netgroups available to the authentication system.

```
    Network> ldap show netgroups
Using ldap-server.example.com as LDAP server and 389 as LDAP port.
=====================
Netgroup 'netgroup1':
=====================
Host                User                Domain
host1               -                   -
host2               admin               -

=====================
Netgroup 'netgroup2':
=====================
Host                User                Domain
host3               testadmin            -
```

Configure the LDAP server name.

```
    Network> ldap set server ldap-server.example.com
OK Completed
```

Get the configured LDAP server name.

```
    Network> ldap get server
LDAP server:              ldap-server.example.com
OK Completed
```

Configure the LDAP server port.

```
    Network> ldap set port 555
OK Completed
```

Get the configured LDAP server port.

```
    Network> ldap get port
LDAP port:                555
OK Completed
```

Configure the base DN.

```
    Network> ldap set basedn dc=example,dc=com
OK Completed
```

Get the configured base DN.

```
    Network> ldap get basedn
LDAP base DN:             dc=example,dc=com
OK Completed
```

Configure LDAP over SSL to on.

```
    Network> ldap set ssl on
OK Completed
```

Get LDAP over SSL setting.

```
    Network> ldap get ssl
LDAP over SSL:            on
OK Completed
```

Configure the bind DN.

```
    Network> ldap set binddn cn=binduser,dc=example,dc=com
Enter password for 'cn=binduser,dc=example,dc=com': ********
OK Completed
```

Get the configured bind DN.

```
    Network> ldap get binddn
LDAP bind DN:             cn=binduser,dc=example,dc=com
OK Completed
```

Configure the root bind DN.

```
    Network> ldap set rootbinddn cn=rootuser,dc=example,dc=com
Enter password for 'cn=rootuser,dc=example,dc=com': ********
OK Completed
```

Get the configured root bind DN.

```
    Network> ldap get binddn
LDAP root bind DN:         cn=rootuser,dc=example,dc=com
OK Completed
```

Configure the password hash.

```
    Network> ldap set password-hash clear
OK Completed
```

Get the configured password hash.

```
    Network> ldap get password-hash
LDAP password hash:        clear
OK Completed
```

Configure the users base DN.

```
    Network> ldap set users-basedn ou=Users,dc=example,dc=com
OK Completed
```

Get the configured users base DN.

```
    Network> ldap get users-basedn
LDAP users base DN:      ou=Users,dc=example,dc=com
OK Completed
```

Configure the groups base DN.

```
    Network> ldap set groups-basedn ou=Groups,dc=example,dc=com
OK Completed
```

Get the configured groups base DN.

```
    Network> ldap get groups-basedn
LDAP groups base DN:      ou=Groups,dc=example,dc=com
OK Completed
```

Configure the netgroups base DN.

```
    Network> ldap set netgroups-basedn ou=Netgroups,dc=example,dc=com
OK Completed
```

Get the configured netgroups base DN.

```
    Network> ldap get netgroups-basedn
LDAP netgroups base DN:    ou=Netgroups,dc=example,dc=com
OK Completed
```

Clear a specified attribute.

```
    Network> ldap clear <attribute>
OK Completed
```

9.8.5 SEE ALSO

bond(1), dns(1), ip(1), netgroup(1), nis(1), nsswitch(1), ping(1), show(1), vlan(1), device(1), swap(1), pciexclusion(1)

9.9 loadbalance

9.9.1 SYNOPSIS

`loadbalance configure` [vip]

`loadbalance status`

`loadbalance remove`

9.9.2 DESCRIPTION

The network `loadbalance` command configures a single VIP to act as a load balancer which distributes the incoming request to the cluster. The command can also remove the load balance configuration.

9.9.3 OPTIONS

`loadbalance configure VIP` Configure the loadbalancer for given VIP.

`loadbalance status` Displays the loadbalance status.

`loadbalance remove` Remove the loadbalance configuration.

9.9.4 EXAMPLES

To configure the loadbalance.

```
Network> loadbalance configure 10.216.114.216
Loadbalancer configured successfully on VIP 10.216.114.216 using node clusternode_01␣
↪as router
```

To display the loadbalance status.

```
Network> loadbalance status

Status               : Enabled
Loadbalance Method   : Round Robin
Router Node          : iplb-beta_03
LoadbalanceIP        : 10.209.193.223

Node                 ActiveConn        InActConn
---------            ----------        ---------
iplb-beta_05         0                 1
iplb-beta_04         1                 0
iplb-beta_03         1                 0
iplb-beta_02         0                 1
iplb-beta_01         1                 0
```

To remove the loadbalance configuration.

```
Network> loadbalance remove
```

9.9.5 SEE ALSO

ipvsadm(1)

9.10 netgroup

9.10.1 SYNOPSIS

`netgroup add` *groupname memberlist*

`netgroup delete` *groupname* [*memberlist*]

`netgroup show` [*groupname*]

9.10.2 DESCRIPTION

The netgroup commands are used to view, add, or delete local netgroups.

The netgroup file defines "netgroups", which are sets of (host, user, domain) tuples, used for permission checking when doing remote mounts, remote logins and remote shells. Each line in the file consists of a netgroup name followed by a list of members, where a member is either another netgroup name, or a triple:

(`host,user,domain`)

where the host, user, and domain are character strings for the corresponding components. Any of the three fields can be empty, in which case specifies a "wildcard", or the string "-" to specify "no valid value". The domain field must either be the local domain name or empty for the netgroup entry to be used. This field does not limit the netgroup or provide security. The domain field refers to the domain in which the triple is valid, not the domain containing the the trusted host.

When exporting a directory by NFS with the specified options, clients may be specified in netgroups. Netgroups may be identified using `@group`. Only the host part of each netgroup member is considered when checking for membership:

```
NFS> share add rw,async /vx/fs1/share @client_group
```

Note, the netgroup takes effect only after adding "files" to the netgroup of Name Service Switch (NSS) configuration:

```
Network> nsswitch conf group files nis
```

When adding data to any existing netgroup, the new data cannot be the same as the existing data of that netgroup. If you are adding data to a non-existent netgroup, you need to create a new netgroup. You can delete the data of an existing netgroup. If you delete a netgroup without a member list, you delete the netgroup.

9.10.3 OPTIONS

groupname netgroup name.

memberlist comma-separated member list. A member can be an IPv4 address, IPv6 address, netgroup name, or a hostname. The hostname can include the characters: a-z| A-Z|0-9 or a hyphen (-). Each level of the hostname should between 1 and 63 characters long and should not start or end with a hyphen (-). The last TLD (Top Level Domain) must be at least two characters and a maximum of six characters. If the member is not a netgroup name, it will be filled in as the host.

`netgroup add` ***groupname memberlist*** Add members to an existing netgroup or create a new netgroup.

`netgroup delete` ***groupname*** **[*memberlist*]** Delete the members of an existing netgroup or delete netgroup.

`netgroup show` **[*groupname*]** Display the netgroup settings.

9.10.4 EXAMPLES

To create a netgroup with members another netgroup name, IPv4 address, IPv6 address, and hostname

```
    Network> netgroup add test_group2 vip,10.200.114.173,cdc.testhost.com,
↪2002:4559:1fe2:0:0:0:45:88

Network> netgroup show
Name                Member List
--------------      -----------
test_group1         192.168.0.8,cdc.myhost.com,2002:4559:1fe2:0:0:0:4559:1f
test_group2         vip,10.200.114.173,cdc.testhost.com,2002:4559:1fe2:0:0:0:45:88
```

To add members to an existing netgroup

```
Network> netgroup add test_group1 10.200.114.250

Network> netgroup show
Name                Member List
--------------      -----------
test_group1         10.200.114.250,192.168.0.8,cdc.myhost.com,
↪2002:4559:1fe2:0:0:0:4559:1f
test_group2         vip,10.200.114.173,cdc.testhost.com,2002:4559:1fe2:0:0:0:45:88
```

To delete members of an existing netgroup

```
    Network> netgroup delete test_group2 2002:4559:1fe2:0:0:0:45:88,cdc.testhost.com,
↪vip

Network> netgroup show
Name                Member List
--------------      -----------
test_group1         10.200.114.250,192.168.0.8,cdc.myhost.com,
↪2002:4559:1fe2:0:0:0:4559:1f
test_group2         10.200.114.173
```

To delete an existing netgroup

```
Network> netgroup delete test_group2

Network> netgroup show
Name                Member List
--------------      -----------
test_group1         10.200.114.250,192.168.0.8,cdc.myhost.com,
↪2002:4559:1fe2:0:0:0:4559:1f
```

Display the netgroup.

```
Network> netgroup show test_group1
Name                Member List
--------------      -----------
test_group1         192.168.0.8,cdc.myhost.com,2002:4559:1fe2:0:0:0:4559:1f
```

9.10.5 SEE ALSO

bond(1), dns(1), ip(1), ldap(1), nis(1), nsswitch(1), ping(1), show(1), vlan(1), device(1), swap(1), pciexclusion(1)

9.11 nis

9.11.1 SYNOPSIS

`nis show` [users|groups|netgroups]

`nis disable`

`nis enable`

`nis set servername` *servername*

`nis set domainname` [*domainname*]

9.11.2 DESCRIPTION

The network `nis` commands set or display the domain name on all of the nodes. The commands can also enable or disable NIS clients on all of the nodes.

9.11.3 OPTIONS

`nis show` Display the NIS server name and domain name.

`nis show users` Show NIS users that are available in the NIS database.

`nis show groups` Show NIS groups that are available in the NIS database.

`nis show netgroups` Show NIS netgroups that are available in the NIS database.

`nis disable` Disable the NIS clients on all the nodes.

`nis enable` Enable NIS clients on all the nodes.

`nis set servername` ***servername*** Set the NIS server name on all the nodes.

`nis set domainname` [***domainname***] Set the NIS domain name on all the nodes.

9.11.4 EXAMPLES

To display the current status of NIS.

```
    Network> nis show
NIS Status    : Disabled
domain        :
NIS Server    :
```

To display NIS netgroups that are available in the NIS database.

```
Network> nis show netgroups
NetGroupname         : netgrp3
Members of the Group : (testqa-13.testqa.com,-,-)
NetGroupname         : netgrp2
Members of the Group : (testqa-13.testqa.com,u2,-),(testqa-13.testqa.com,u1,-)
```

To display NIS groups that are available in the NIS database.

```
Network> nis show groups
Groupname            : webservd
Members of the Group :
Groupname            : sysadmin
Members of the Group :
Groupname            : noaccess
Members of the Group :
```

To display NIS users that are available in the NIS database.

```
Network> nis show users
User           : root
Name           : Super-User
uid            : 0
gid            : 0
Home Directory : /
Login Shell    : /sbin/sh

User           : u2
Name           :
uid            : 101
gid            : 1
Home Directory : /lhome/u2
Login Shell    : /bin/sh
```

To set the domain name.

```
Network> nis set domainname vxindia.veritas.com
Setting domainname: "vxindia.veritas.com"
```

To set the server name.

```
Network> nis set servername vmlxpx1.vxindia.veritas.com
Setting NIS Server "vmlxpx1.vxindia.veritas.com"
```

To enable NIS.

```
Network> nis enable
```

To check whether NIS is enabled or not, run the following command:

```
Network> nis show
NIS Status    : Enabled
domain        : vxindia.veritas.com
NIS Server    : vmlxpx1.vxindia.veritas.com
```

To disable the NIS.

```
Network> nis disable
```

9.11.5 SEE ALSO

bond(1), dns(1), ip(1), ldap(1), netgroup(1), nsswitch(1), ping(1), show(1), vlan(1), device(1), swap(1), pciexclusion(1)

9.12 nsswitch

9.12.1 SYNOPSIS

`nsswitch show`

`nsswitch conf` { group|hosts|netgroup|passwd|shadow } *value1 value2 value*

9.12.2 DESCRIPTION

The network `nsswitch` commands display or change the Name Service Switch (NSS) configuration.

9.12.3 OPTIONS

`nsswitch show` Display the name service switch configuration.

`nsswitch conf` **{ group|hosts|netgroup|passwd|shadow }** ***value1*** **[*value2*] [*value3*] [*value4*]**

Configure the method for the name service switch lookup process. Specify the name service switch lookup order with the following values:

value1: Choose the type {files} {files}

value2: Enter one of the following types { files | nis | winbind | ldap }

value3: Enter one of the following types { files | nis | winbind | ldap }

value4: Enter one of the following types { files | nis | winbind | ldap }

To configure `nsswitch conf host` *value1* [*value2*] [*value3*] enter the following:

value1: Choose the type {files} {files}

value2: Enter one of the following types { files | nis | dns }

value3: Enter one of the following types { files | nis | dns }

To configure `nsswitch conf netgroup` *value1* [*value2*] [*value3*], enter the following:

value1: Enter one of the following types { files | nis | winbind | ldap }

value2: Enter one of the following types { files | nis | winbind | ldap }

9.12.4 EXAMPLES

To display the current value set on nsswitch for all group, hosts, netgroup, passwd and shadow

```
Network> nsswitch show
group:      files     nis winbind    ldap
hosts:      files     nis            dns
netgroup:   files     nis
passwd:     files     nis winbind    ldap
shadow:     files     nis winbind
```

Change the order of the group items.

```
Network> nsswitch conf group nis files
```

Immediate effect of above command can be seen by using the show command.

```
Network> nsswitch show
 group:    files     nis
 hosts:    files     nis            dns
 netgroup: nis       files
 passwd:   files     nis winbind    ldap
 shadow:   files     nis winbind
```

To configure netgroup

```
Network> nsswitch conf netgroup files ldap nis
```

To configure passwd

```
Network> nsswitch conf passwd files winbind ldap nis
```

To configure shadow

```
Network> nsswitch conf shadow files winbind
```

9.12.5 SEE ALSO

bond(1), dns(1), ip(1), nis(1), ldap(1), netgroup(1), ping(1), show(1), vlan(1), device(1), swap(1), pciexclusion(1)

9.13 pciexclusion

9.13.1 SYNOPSIS

`pciexclusion show`

`pciexclusion add` *pcilist*

`pciexclusion delete` *pci*

9.13.2 OPTIONS

pcilist This includes comma-separated list of PCI IDs. The PCI ID bits format is hexadecimal. (XXXX:XX:XX.X).

pci The PCI ID bits format is hexadecimal (XXXX:XX:XX.X).

`pciexclusion show` Displays the PCI IDs that have been selected for exclusion. It also provides information about whether it has been excluded or not by displaying y(yes) or n(no) symbols corresponding to the node name. If the node is in the INSTALLED state, it displays the UUID of the node.

`pciexclusion add` ***pcilist*** Allows an administrator to add specific PCI ID(s) for exclusion. These values must be provided before doing the installation. It then excludes the PCI from the second node installation.

`pciexclusion delete` ***pci*** Allows an administrator to delete a given PCI ID from exclusion. This command must be used before doing the installation for it to take effect.

9.13.3 EXAMPLES

Display the PCI IDs.

```
    Support> pciexclusion show

PCI ID              EXCLUDED      NODENAME/UUID
------              --------      -------------
0000:0e:00.0         y             test_1
0000:0e:00.0         y             a79a7f43-9fe2-4eeb-aa1f-27a70e7a0820
0000:04:00:1         n
```

Add the PCI ID for exclusion.

```
    Support> pciexclusion add 0000:00:09.0
    Support> pciexclusion show

PCI ID              EXCLUDED      NODENAME/UUID
------              --------      -------------
0000:0e:00.0         y             test_1
0000:0e:00.0         y             a79a7f43-9fe2-4eeb-aa1f-27a70e7a0820
0000:04:00:1         n
0000:00:09.0         n
```

Delete the PCI ID from exclusion.

```
    Support> pciexclusion delete 0000:04:00:1
    Support> pciexclusion show

PCI ID              EXCLUDED      NODENAME/UUID
------              --------      -------------
0000:0e:00.0         y             test_1
0000:0e:00.0         y             a79a7f43-9fe2-4eeb-aa1f-27a70e7a0820
0000:00:09.0         n
```

9.13.4 SEE ALSO

bond(1), dns(1), ip(1), ldap(1), netgroup(1), nis(1), nsswitch(1), ping(1), show(1), vlan(1), device(1), swap(1)

9.14 ping

9.14.1 SYNOPSIS

`ping` *destination* [*nodename*] [*devicename*] [*packets*]

9.14.2 DESCRIPTION

The network `ping` command is used to test whether a particular host or gateway is reachable across an IP network.

9.14.3 OPTIONS

destination The host or gateway to send the information to. The destination field can contain either a DNS name or an IP address.

nodename Node from which the ping takes place. A value of `any` indicates the ping can be done from any node in the cluster.

devicename Device through which the ping takes place. A value of `any` indicates the ping can be done from any device in the cluster.

packets Number of packets to be sent to the destination. The default is 5.

9.14.4 EXAMPLES

Test whether *destination* is reachable across an IP network.

```
Network> ping somenode
PING somenode.domain.com (10.216.2.11) 56(84) bytes of data.
64 bytes from somenode.domain.com (10.216.2.11): icmp_seq=1 ttl=64 time=0.134 ms
64 bytes from somenode.domain.com (10.216.2.11): icmp_seq=2 ttl=64 time=0.154 ms
64 bytes from somenode.domain.com (10.216.2.11): icmp_seq=3 ttl=64 time=0.167 ms
64 bytes from somenode.domain.com (10.216.2.11): icmp_seq=4 ttl=64 time=0.114 ms
64 bytes from somenode.domain.com (10.216.2.11): icmp_seq=5 ttl=64 time=0.160 ms


--- somenode.domain.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.114/0.145/0.167/0.024 ms
```

Test whether *destination* is reachable across an IP network from *nodename*.

```
Network> ping somenode node1_1
PING somenode.domain.com (10.216.2.11) 56(84) bytes of data.
64 bytes from somenode.domain.com (10.216.2.11): icmp_seq=1 ttl=64 time=0.134 ms
64 bytes from somenode.domain.com (10.216.2.11): icmp_seq=2 ttl=64 time=0.154 ms
64 bytes from somenode.domain.com (10.216.2.11): icmp_seq=3 ttl=64 time=0.167 ms
64 bytes from somenode.domain.com (10.216.2.11): icmp_seq=4 ttl=64 time=0.114 ms
64 bytes from somenode.domain.com (10.216.2.11): icmp_seq=5 ttl=64 time=0.160 ms


--- somenode.domain.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.114/0.145/0.167/0.024 ms
```

Test whether *destination* is reachable across an IP network from *nodename*, via a specific *devicename*.

```
Network> ping somenode node1_1 pubeth0
PING somenode.domain.com (10.216.2.11) 56(84) bytes of data.
64 bytes from somenode.domain.com (10.216.2.11): icmp_seq=1 ttl=64 time=0.134 ms
64 bytes from somenode.domain.com (10.216.2.11): icmp_seq=2 ttl=64 time=0.154 ms
64 bytes from somenode.domain.com (10.216.2.11): icmp_seq=1 ttl=64 time=0.134 ms
64 bytes from somenode.domain.com (10.216.2.11): icmp_seq=2 ttl=64 time=0.154 ms
64 bytes from somenode.domain.com (10.216.2.11): icmp_seq=2 ttl=64 time=0.154 ms

--- somenode.domain.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4000ms
rtt min/avg/max/mdev = 0.018/0.026/0.057/0.016 ms
```

Test whether *destination* is reachable across an IP network from *nodename*, by sending a specified number of packets.

```
Network> ping somenode node1_1 any 2
PING somenode.domain.com (10.216.2.11) 56(84) bytes of data.
64 bytes from somenode.domain.com (10.216.2.11): icmp_seq=1 ttl=64 time=0.134 ms
64 bytes from somenode.domain.com (10.216.2.11): icmp_seq=2 ttl=64 time=0.154 ms


--- somenode.domain.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.114/0.145/0.167/0.024 ms
```

9.14.5 SEE ALSO

bond(1), dns(1), ip(1), ldap(1), netgroup(1), nis(1), nsswitch(1), ping(1), show(1), vlan(1), device(1), swap(1), pciexclusion(1)

9.15 show

9.15.1 SYNOPSIS

`show`

9.15.2 DESCRIPTION

The network `show` command displays the network configuration and statistics on all of the available nodes.

9.15.3 EXAMPLES

Show the network configuration and statistics.

```
va73> network show

Interface Statistics
--------------------

va73_01
-------------
Interfaces         MTU       RX-OK     RX-DROP      RX-ERR    RX-FRAME       TX-OK
↪ TX-DROP      TX-ERR      TX-CAR         Flag
        lo       65536       48138           0           0           0       48138
↪        0           0           0          LU
  priveth0        1500      955874          18           0           0      912458
↪        0           0           0        BMRU
  priveth1        1500      759216          18           0           0      647319
↪        0           0           0        BMRU
   pubeth0        1500     1757268         794           0           0       82759
↪        0           0           0        BMRU


va73_02
-------------
Interfaces         MTU       RX-OK     RX-DROP      RX-ERR    RX-FRAME       TX-OK
↪ TX-DROP      TX-ERR      TX-CAR         Flag
        lo       65536       27296           0           0           0       27296
↪        0           0           0          LU
  priveth0        1500     1062983          14           0           0      805435
↪        0           0           0        BMRU
  priveth1        1500      797769          14           0           0      608673
↪        0           0           0        BMRU
   pubeth0        1500     1761159         819           0           0       85103
↪        0           0           0        BMRU


Routing Table
-------------

va73_01
-------------
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         10.209.192.1    0.0.0.0         UG        0 0          0 pubeth0
10.209.192.0    0.0.0.0         255.255.252.0   U         0 0          0 pubeth0
```

```
172.16.0.0      0.0.0.0         255.255.255.0   U         0 0          0 priveth0


va73_02
-------------
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         10.209.192.1    0.0.0.0         UG        0 0          0 pubeth0
10.209.192.0    0.0.0.0         255.255.252.0   U         0 0          0 pubeth0
172.16.0.0      0.0.0.0         255.255.255.0   U         0 0          0 priveth0
```

9.15.4 OUTPUT

Interface Statistics

Interfaces The name of the interface.

MTU The Maximum Transmission Unit of the interface.

RX-OK Number of received packets.

RX-DROP Number of dropped packets.

RX-ERR Number of error packets.

RX-FRAME Number of packets fails to end on byte boundary.

TX-OK Number of transmitted packets.

TX-DROP Number of dropped packets.

TX-ERR Number of error packets.

TX-CAR Number of carrier lost.

Flag

B: BROADCAST L: LOOPBACK M: MULTICAST R: RUNNING U: UP

Routing Table

Destination The destination network or destination host.

Gateway The gateway address or '*' if none set.

Genmask The netmask for the destination net; '255.255.255.255' for a host destination and '0.0.0.0' for the default route.

Flags U (route is up) H (target is a host) G (use gateway) R (reinstate route for dynamic routing) D (dynamically installed by daemon or redirect) M (modified from routing daemon or redirect) A (installed by addrconf) C (cache entry) ! (reject route)

Next Hop The the nexthop router of a multipath route.

Ref Number of references to this route.

Use Count of lookups for the route.

MSS Default maximum segment size for TCP connections over this route.

Window Default window size for TCP connections over this route.

irtt Initial RTT (Round Trip Time).

Iface Interface to which packets for this route will be sent.

9.15.5 SEE ALSO

bond(1), dns(1), ip(1), ldap(1), netgroup(1), nis(1), nsswitch(1), ping(1), vlan(1), device(1), swap(1), pciexclusion(1)

9.16 swap

9.16.1 SYNOPSIS

`swap` *interface1 interface2* [*nodename*]

9.16.2 DESCRIPTION

The system `swap` command can be used for swapping two network interfaces of a node in a cluster. The System Administrator can use this command with a multi-node cluster only to swap public interfaces. On single node cluster, this command can be used to swap public as well as private interfaces. If input to the `swap` command contains one public and one private interface, and there are two separate switches for the private and the public network, then before running this command, the System Administrator has to exchange cable connections between these interfaces. This command requires stopping the given interfaces, and after running the `swap` command, all SSH connection(s) hosted on the input interfaces will terminate. The `swap` command helps set up the cluster properly in cases where the installation of the first node gateway cannot be pinged and is not to be used with a full-fledged system. It is not recommended that you use this command when you have some CIFS/NFS shares exported. You can check the status of the `swap` command under `history`.

9.16.3 OPTIONS

`swap` ***interface1 interface2 nodename*** Swap two network interfaces, interface1 and interface2.

interface1 : name of first interface.

interface2 : name of second interface.

nodename : name of node. If nodename is not provided, it is executed on the current node.

9.16.4 EXAMPLES

Swap two interfaces.

```
    Network> swap pubeth0 priveth0 test_01
All ssh connection(s) to swapped interfaces need to start again after this command.
Do you want to continue [Enter "y/yes" to continue]...
```

Check status of this command in history.

9.16.5 SEE ALSO

bond(1), dns(1), ip(1), ldap(1), netgroup(1), nis(1), nsswitch(1), ping(1), show(1), vlan(1), device(1), pciexclusion(1)

9.17 vlan

9.17.1 SYNOPSIS

`vlan show`

`vlan add` *device vlan_id*

`vlan del` *vlan_device*

9.17.2 DESCRIPTION

The network `vlan` command is used to view, add, or delete VLAN interfaces.

By default `ip addr add` will not use VLAN devices unless explicity specified in the *device* attribute. By default, CIFS shares created using `share add` will not use virtual IPs based on VLAN devices, unless explicitly specified with *ip=* in share options.

When a VLAN device is deleted, physical IPs that are used by that VLAN device will automatically be freed, and can be seen in the output of `ip addr show` as unused. When a VLAN device is deleted, virtual IPs configured on that VLAN device will automatically be deleted and will no longer be visible in `ip addr show`.

9.17.3 OPTIONS

device Device on which VLAN devices needs to be added.

vlan_id VLAN id which the new VLAN device should use. Valid values range from 1-4095.

vlan_device A VLAN device name is a combination of actual device on which the VLAN is based and the VLAN id separated by '.'.

`vlan show` Show the current VLAN devices.

`vlan add` ***device vlan_id*** Add a VLAN device on *device* with VLAN id as fIvlan_id.

`vlan del` ***vlan_device*** Delete the VLAN device named *vlan_device*.

9.17.4 EXAMPLES

Display the VLAN devices.

```
Network> vlan show
VLAN            DEVICE     VLAN id
----            ------     -------
pubeth0.2       pubeth0    2
```

To add a VLAN device with id 2 on pubeth1

```
Network> vlan add pubeth1 2

Network> vlan show
VLAN            DEVICE     VLAN id
----            ------     -------
pubeth0.2       pubeth0    2
pubeth1.2       pubeth1    2
```

To delete a VLAN device that is shown in the output of *vlan show*

```
Network> vlan del pubeth0.2

Network> vlan show
VLAN            DEVICE     VLAN id
----            ------     -------
pubeth1.2       pubeth1    2
```

9.17.5 SEE ALSO

bond(1), dns(1), ip(1), ldap(1), netgroup(1), nis(1), nsswitch(1), ping(1), show(1), device(1), swap(1), pciexclusion(1)

10

NFS Commands

10.1 nfs

10.1.1 SYNOPSIS

`server` [start|stop|status|switch]

`stat show` [*nodename*]

`stat reset` [*nodename*]

`share add` *nfsoptions export_dir* [*client*]

`share delete` *export_dir* [*client*]

`share show`

`show fs`

10.1.2 DESCRIPTION

The `nfs` commands maintain the current table of exported file systems for the Network File System (NFS).

10.1.3 OPTIONS

nfsoptions Comma-separated list of export options from the set {ro, rw, sync, async, secure, insecure, secure_locks, insecure_locks, root_squash, no_root_squash, wdelay, no_wdelay, subtree_check, no_subtree_check, nordirplus, fsid=<number>, sec=krb5:krb5i:krb5p:sys:none}. For example: ro,root_squash; rw,secure,no_root_squash. The default export options are {ro, sync, secure, secure_locks, root_squash, wdelay, no_subtree_check}. The "no_wdelay" option has no effect if "async" is also set. fsid could be any number between 1 and 1073741823 and it must be unique among all exported paths.

wdelay: For backwards-compatibility, the wdelay and no_wdelay options are still supported, but the options are being deprecated. no_wdelay is the the only effective behavior regardless of setting wdelay or no_wdelay option. Clients can handle flushing of data using server COMMIT operations and UNSTABLE writes.

For NFS Ganesha exports, export options {secure_locks, insecure_locks, wdelay, no_wdelay, subtree_check, no_subtree_check, fsid=number} are not supported and will be ignored.

export_dir Path of the directory that needs to exported to the client. Path should start with /vx and only "a-zA-Z0-9_/.:-" characters are allowed in *export_dir*.

`server` [start|stop|status|switch] Start, stop, or check the status of NFS resources. Server switch command will toggle active server from kernel NFS server (KNFS) to usermode Ganesha NFS server (GNFS) and vice versa.

`stat show` [*nodename*] Display NFS statistics for *nodename* or all the nodes in the cluster.

`stat reset` [*nodename*] Reset NFS statistics for *nodename* or all the nodes in the cluster to zero. Statistics are automatically reset to zero after reboot of a node. This command is not supported with GNFS.

`share add` *nfsoptions export_dir* [*client*] Export the directory with the specified options. Clients may be specified in the following ways: 1. Single host

Specify a host either by an abbreviated name recognized by the resolver, the fully qualified domain name, or an IP address.

2. Netgroups

Netgroups may be given as @*group*. Only the host part of each netgroup member is considered when checking for membership.

3. IP networks

You can also simultaneously export directories to all hosts on an IP (sub-) network. This is done by specifying an IP address and netmask pair as *address/netmask* where the netmask can be specified as a contiguous mask length. IPv4 or IPv6 addresses can be used.

If the *client* is not given, then the specified file system can be accessed or mounted by any client.

To re-export new options to an existing share, the new options will be updated after the command is run.

`share delete` *export_dir* [*client*] Unexport the directory.

`share show` List all of the exported resources.

`show fs` List all of the online file systems and snapshots that can be exported.

10.1.4 EXAMPLES

Start the NFS server. If the NFS server is already started, the start command clears any faults and then tries to start the NFS server.

```
NFS> server start
..Success.
```

Display the status of the NFS server.

```
NFS> server status
NFS Status on nas_01 : ONLINE
NFS Status on nas_02 : ONLINE
```

Display the status of the Ganesha NFS server.

```
NFS> server status
GNFS Status on nas_01 : ONLINE
GNFS Status on nas_02 : ONLINE
```

Stop the NFS server. You will receive an error if you try to stop an already stopped NFS server.

```
NFS> server stop
..Success.
```

Display the list of FS/snapshot.

```
NFS> show fs
FS/Snapshot
===========
fs1
```

Export the directory `/vx/fs1` to every host allowing asynchronous writes.

```
NFS> share add rw,async /vx/fs1
Exporting \*:/vx/fs1 with options rw,async
..Success.
```

Export the directory `/vx/fs1` using the krb5 security option.

```
NFS> share add sec=krb5 /vx/fs1
Exporting /vx/fs1 with options sec=krb5
Success.
```

Export a directory using krb5, krb5i, krb5p, and sys options. Different clients can use different levels of security in this case. Client A can mount with krb5, and client B can mount with krb5p. If no mount option is given at the client side, security to be chosen is negotiated, and the highest level of security is chosen. In this case, krb5p.

```
NFS> share add sec=krb5:krb5i:krb5p:sys /vx/fs1
Exporting /vx/fs1 with options sec=krb5:krb5i:krb5p:sys
Success.
```

Show the exported file system for KNFS.

```
NFS> share show
/vx/fs1                              * (rw,async)
```

Unexport the directory `/vx/fs1` from world.

```
NFS> share delete /vx/fs1
Removing export path \*:/vx/fs1
..Success.
```

10.1.5 SEE ALSO

server(1), share(1), stat(1)

10.2 server

10.2.1 SYNOPSIS

`server` [start|stop|status|switch]

10.2.2 DESCRIPTION

The NFS `server` command starts, stops, switches, or checks the status of the NFS resources.

10.2.3 OPTIONS

`server` [start|stop|status|switch] Start, stop, or check the status of NFS resources. Server switch command will toggle active server from the kernel NFS server (KNFS) to usermode NFS-Ganesha (GNFS) or vice versa.

In case of the GNFS server, `start` command checks for the correct domain in the /etc/idmapd.conf file. If the domain is not set, NFSv4 ID mapping is not correct, and the command throws a warning to set the domain.

10.2.4 EXAMPLES

Start the NFS server. If the NFS server is already started, Veritas Access clears the faults (if any), and then tries to start the NFS server.

```
NFS> server start
..Success.
```

Display the status of the NFS server.

```
NFS> server status
NFS Status on nas_01 : ONLINE
NFS Status on nas_02 : ONLINE
```

Display the status of the Ganesha NFS server.

```
NFS> server status
GNFS Status on nas_01 : ONLINE
GNFS Status on nas_02 : ONLINE
```

Stop the NFS server. You will receive an error if you try to stop an already stopped NFS server.

```
NFS> server stop
..Success.
```

Switch the NFS server. You will receive an error if the NFS server is running.

```
NFS> server switch
This command will switch NFS server from GNFS to KNFS and will take time.
Do you want to continue (y/n): y
ACCESS nfs SUCCESS V-288-0 NFS server switch from GNFS to NFS successful. Please␣
↪start NFS service.
```

10.2.5 SEE ALSO

share(1), stat(1)

10.3 share

10.3.1 SYNOPSIS

`share add` *nfsoptions export_dir* [*client*]

`share delete` *export_dir* [*client*]

`share show`

10.3.2 DESCRIPTION

The NFS `share` commands add, delete, or display the NFS share.

10.3.3 OPTIONS

nfsoptions Comma-separated list of export options from the set {ro, rw, sync, async, secure, insecure, secure_locks, insecure_locks, root_squash, no_root_squash, wdelay, no_wdelay, subtree_check, no_subtree_check, nordirplus, fsid=<number>, sec=krb5:krb5i:krb5p:sys:none}. For example: ro,root_squash; rw,secure,no_root_squash. The default export options are {ro, sync, secure, secure_locks, root_squash, wdelay, no_subtree_check}. The "no_wdelay" option has no effect if "async" is also set. fsid could be any number between 1 and 1073741823. fsid must be unique among all exported paths.

wdelay: For backwards-compatibility, the wdelay and no_wdelay options are still supported, but the options are being deprecated. The effective behavior is no_wdelay is used, even when wdelay is set. Clients can handle flushing of data using server COMMIT operations and UNSTABLE writes.

For NFS-Ganesha exports, export options {secure_locks, insecure_locks, wdelay, no_wdelay, subtree_check, no_subtree_check, fsid=number} are not supported and will be ignored.

export_dir Path of the directory that needs to exported to the client. The path should start with /vx and only 'a-zA-Z0-9_/.:-' characters are allowed in *export_dir*.

`share add` *nfsoptions export_dir* [*client*] Export the directory with the specified options. Clients may be specified in the following ways:

1. Single host

   Specify a host either by an abbreviated name recognized by the resolver, the fully qualified domain name, or an IP address.

2. Netgroups

   Netgroups may be given as @*group*. Only the host part of each netgroup member is considered when checking for membership.

3. IP networks

   You can also simultaneously export directories to all hosts on an IP (sub-) network. This is done by specifying an IP address and netmask pair as *address/netmask* where the netmask can be specified as a contiguous mask length. IPv4 addresses or IPv6 addresses can be used.

If the *client* is not given, then the specified file system can be accessed or mounted by any client.

To re-export new options to an existing share, the new options will be updated after the command is run.

`share delete` *export_dir* [*client*] Unexport the directory.

share show List all of the exported resources. You can use any of the VIPs to mount the NFS shares from the NFS client.

10.3.4 EXAMPLES

Export the directory `/vx/fs1` to every host allowing asynchronous writes.

```
NFS> share add rw,async /vx/fs1
Exporting \*:/vx/fs1 with options rw,async
..Success.
```

Export the directory `/vx/fs1` using the krb5 security option. This is allowed only for GNFS.

```
NFS> share add sec=krb5 /vx/fs1
Exporting /vx/fs1 with options sec=krb5
Success.
```

Export the directory `/vx/fs1` using krb5, krb5i, krb5p, and sys options. This is allowed only for GNFS. Different clients can use different levels of security in this case. Client A can mount with krb5, and client B can mount with krb5p. If no mount option is given at the client side, security to be chosen is negotiated, and the highest level of security is chosen. In this case, krb5p.

```
NFS> share add sec=krb5:krb5i:krb5p:sys /vx/fs1
Exporting /vx/fs1 with options sec=krb5:krb5i:krb5p:sys
Success.
```

Show the exported file system for KNFS.

```
NFS> share show
/vx/fs1                              * (rw,async)
```

Show the exported file system for GNFS. In the following example, IP 10.209.106.180 should be used to mount the share on the client if fs1 is largefs.

```
NFS> share show
/vx/fs1                              * (rw,async) 10.209.106.180
```

Unexport the directory `/vx/fs1` from world.

```
NFS> share delete /vx/fs1
Removing export path \*:/vx/fs1
..Success.
```

10.3.5 SEE ALSO

server(1), stat(1)

10.4 show

10.4.1 SYNOPSIS

`show fs`

10.4.2 DESCRIPTION

The `show fs` command displays a list of all online file systems and snapshots that can be exported.

10.4.3 EXAMPLES

Display the list of file systems and snapshots.

```
NFS> show fs
FS/Snapshot
===========
fs1
```

10.4.4 SEE ALSO

share(1), server(1)

10.5 stat

10.5.1 SYNOPSIS

`stat show` [*nodename*]

`stat reset` [*nodename*]

10.5.2 DESCRIPTION

The NFS `stat` manages NFS statistics.

10.5.3 OPTIONS

`stat show` [*nodename*] Display NFS statistics for *nodename* or all nodes in the cluster.

`stat reset` [*nodename*] Reset NFS statistics for *nodename* or all the nodes in the cluster to zero. Statistics are automatically reset to zero after reboot of a node. This command is not supported with GNFS.

10.5.4 EXAMPLES

Show NFS statistics for all nodes in the cluster.

```
NFS> stat show all

node_01
-------
Server rpc stats:
calls       badcalls        badclnt         badauth         xdrcall
    0               0                       0                       0
↪       0

Server nfs v3:
null         getattr      setattr      lookup       access       readlink
    0 0%             0 0%         0 0%         0 0%         0 0%         0 0%
read         write        create       mkdir        symlink      mknod
    0 0%             0 0%         0 0%         0 0%         0 0%         0 0%
remove       rmdir        rename       link         readdir      readdirplus
    0 0%             0 0%         0 0%         0 0%         0 0%         0 0%
fsstat       fsinfo       pathconf     commit
    0 0%             0 0%         0 0%         0 0%

    Server nfs v4:
    null    compound
    0 0%    0 0%

    Server nfs v4 operations:
    op0-unused              op1-unused              op2-future      access
↪       close           commit
    0 0%                    0 0%                    0 0%            0 0%
↪       0 0%            0 0%
    create                  delegpurge              delegreturn getattr
↪getfh           link
```

```
    0 0%                    0 0%                    0 0%            0 0%
↪       0 0%            0 0%
    lock                    lockt                   locku           lookup
↪       lookup_root nverify
    0 0%                    0 0%                    0 0%            0 0%
↪       0 0%            0 0%
    open                    openattr                open_conf       open_dgrd
↪       putfh           putpubfh
    0 0%                    0 0%                    0 0%            0 0%
↪       0 0%            0 0%
    putrootfh               read                    readdir         readlink
↪       remove          rename
    0 0%                    0 0%                    0 0%            0 0%
↪       0 0%            0 0%
    renew                   restorefh               savefh          secinfo
↪       setattr         setcltid
    0 0%                    0 0%                    0 0%            0 0%
↪       0 0%            0 0%
    setcltid                confverify              write           rellockowner
↪bc_ctl          bind_conn
    0 0%                    0 0%                    0 0%            0 0%
↪       0 0%            0 0%
    exchange_id     create_ses              destroy_ses free_stateid
↪getdirdeleg getdevinfo
    0 0%                    0 0%                    0 0%            0 0%
↪       0 0%            0 0%
    getdevlist              layoutcommit    layoutget       layoutreturn
↪secinfononam sequence
    0 0%                    0 0%                    0 0%            0 0%
↪       0 0%            0 0%
    set_ssv                 test_stateid    want_deleg      destroy_clid    reclaim_
↪comp
    0 0%                    0 0%                    0 0%            0 0%
↪       0 0%

node_02
-------
Server rpc stats:
calls       badcalls        badclnt badauth xdrcall
    0               0                       0               0               0

Server nfs v3:
null         getattr      setattr      lookup       access       readlink
    0 0%             0 0%             0 0%             0 0%                 0 0%
↪       0 0%
read         write        create       mkdir        symlink      mknod
    0 0%             0 0%             0 0%             0 0%                 0 0%
↪       0 0%
remove       rmdir        rename       link         readdir      readdirplus
    0 0%             0 0%             0 0%             0 0%                 0 0%
↪       0 0%
fsstat       fsinfo       pathconf     commit
    0 0%             0 0%             0 0%             0 0%

    Server nfs v4:
    null    compound
    0 0%    0 0%
```

```
    Server nfs v4 operations:
    op0-unused              op1-unused              op2-future      access          ␣
↪       close           commit
    0 0%                    0 0%                    0 0%            0 0%            ␣
↪       0 0%            0 0%
    create                  delegpurge              delegreturn getattr             ␣
↪getfh           link
    0 0%                    0 0%                    0 0%            0 0%            ␣
↪       0 0%            0 0%
    lock                    lockt                   locku           lookup          ␣
↪       lookup_root nverify
    0 0%                    0 0%                    0 0%            0 0%            ␣
↪       0 0%            0 0%
    open                    openattr                open_conf       open_dgrd       ␣
↪       putfh           putpubfh
    0 0%                    0 0%                    0 0%            0 0%            ␣
↪       0 0%            0 0%
    putrootfh               read                    readdir         readlink        ␣
↪       remove          rename
    0 0%                    0 0%                    0 0%            0 0%            ␣
↪       0 0%            0 0%
    renew                   restorefh               savefh          secinfo         ␣
↪       setattr         setcltid
    0 0%                    0 0%                    0 0%            0 0%            ␣
↪       0 0%            0 0%
    setcltid                confverify              write           rellockowner    ␣
↪bc_ctl          bind_conn
    0 0%                    0 0%                    0 0%            0 0%            ␣
↪       0 0%            0 0%
    exchange_id     create_ses              destroy_ses free_stateid        ␣
↪getdirdeleg getdevinfo
    0 0%                    0 0%                    0 0%            0 0%            ␣
↪       0 0%            0 0%
    getdevlist              layoutcommit    layoutget       layoutreturn    ␣
↪secinfononam sequence
    0 0%                    0 0%                    0 0%            0 0%            ␣
↪       0 0%            0 0%
    set_ssv                 test_stateid    want_deleg      destroy_clid    reclaim_
↪comp
    0 0%                    0 0%                    0 0%            0 0%            ␣
↪       0 0%


NFS>
```

Show GNFS statistics for all the nodes in the cluster.

```
NFS> stat show all

node_01
-------
EXPORT PATH     PROTO    TYPE       REQ (MB)   XFER (MB)   TOTAL OPS AVG LATENCY (ms)
/vx/fs1         NFSv3    READ           0.00        0.00           0        0.00
/vx/fs1         NFSv3    WRITE         48.00       48.00          48       11.85
/vx/fs1         NFSv4    READ           0.00        0.00           0        0.00
/vx/fs1         NFSv4    WRITE        100.00      100.00         100        8.42

node_02
```

```
-------
EXPORT PATH     PROTO    TYPE       REQ (MB)   XFER (MB)   TOTAL OPS AVG LATENCY (ms)
/vx/fs1         NFSv3    READ           1.00        0.00           0        0.00
/vx/fs1         NFSv3    WRITE         51.00       78.00          53       17.85
/vx/fs1         NFSv4    READ           9.00        0.00           0        0.00
/vx/fs1         NFSv4    WRITE        108.00      116.00         111        9.42

NFS>
```

Reset NFS statistics to zero on node_02.

```
NFS> stat reset node_02
Success.
```

10.5.5 SEE ALSO

share(1), server(1)

11

Object Access Commands

11.1 ObjectAccess

11.1.1 SYNOPSIS

`account` user show

`bucket` show [bucket_name]

`group` [set|show|unset]

`server` [enable|start|stop|status|disable]

`set` [ssl_enabled|pools|fs_size|fs_type]

`unset` pools pool1[,pool2,...]

`map` filesystem_path username

`show`

11.1.2 DESCRIPTION

The ObjectAccess commands are used to manage the object access service, to show the status of the service, to set and unset the default or group specific configuration, and to manage the buckets accessed using the Amazon S3 protocol.

The following are the steps to configure and use the Veritas object access service.

1. Set up the default runtime configuration options. You need to set at least one default pool. For example, `objectaccess set pools slow_pool`.
2. Enable the object access server.
3. Start the object access server.

The object access server creates a new file system for every new bucket. The type, size, and pools for the new file system should be configured correctly by using the object access `set` command. If the `group set fs_sharing` option is set to no, a new file system is created for the bucket every time. If the `group set fs_sharing` option is set to yes, then new buckets are created under the existing file system. If the administrator expects a large number of buckets to be created, the administrator can choose to share the same file system across all the users of the given group.

See the `objectaccess_group man` page for more information.

You cannot start the object access service until you have enabled it.

4. Configure NIS/LDAP/AD with Veritas Access and create keys for user authentication.

The object access server provides a custom REST-based API to create an access key and a secret key. The REST APIs use password-based authentication. It is recommended to always set `ssl_enabled` for the server. See the *Veritas Access RESTful API Guide* for more information.

5. Set up user groups.

The object access server can be configured to use specific configurations based on the requester user's group. An administrator can override the file system type, size, pools, which should be used for a particular group. These group-specific options override the default options. For users that are not part of any pre-configured object access groups, the server uses the default configuration to create the file system.

Consider a pool, slow_pool, is created consisting of HDDs, and slow_pool is configured as the default pool. The default pool, slow_pool, applies to all users.

Consider a pool, fast_pool, is created consisting of only SSDs, and you want some users to use fast_pool for creating their buckets instead of the default slow_pool, then use the `group` options.

6. Use the object access server using an S3-API compatible client.

11.1.3 OPTIONS

`account` user [show] List the S3 users.

`bucket` show [bucket_name] List the buckets created by the S3 users.

group [set|show|unset] Groups allow an administrator to override the default file system configuration parameters. This gives fine control over how the object access server creates file systems and buckets for particular user groups. It is recommended to create new groups using the authentication server (NIS/LDAP/AD) based on your storage requirements and use them for this group configuration.

`server` [enable|start|stop|status|disable] Start, stop, or check the status of the object access service. The server must be enabled before starting. Disabling the server deletes the object access configuration and user information database. All S3 buckets must be deleted before disabling the object access server.

`set` [ssl_enabled|pools|fs_size|fs_type] Set object access configurable and default file system options. The object access server creates a new file system for every new bucket. Use the `set` command to create a new file system of a different configuration. The `ssl_enabled` option enables HTTPS access. The `pools` option lets you choose the pool on which the bucket has to be created. Use the `fs_size` option to set the default fs_size for the buckets.

`unset` pools pool1[,pool2,...] Remove pools from the default pool list. Removing all pools is not allowed as object access requires at least one default pool.

`show` List all the object access configurable options.

`map` filesystem_path username Map filesytem path as a bucket for a particular user. If filesystem path is not present then map command would create the directory and set the default S3 bucket permissions. The ownership to the bucket will be set to the user and it's group. Filesystem path should not be in a directory inside exported nfs share or inside existing bucket. Buket name will be same as a directory name.

11.1.4 EXAMPLES

Enable the object access service.

```
earth.ObjectAccess> server enable
100% [********************] Enabling ObjectAccess server.
ACCESS ObjectAccess SUCCESS V-288-0 ObjectAccess server enabled.
```

Start the object access service.

```
earth.ObjectAccess> server start
ACCESS ObjectAccess SUCCESS V-288-0 ObjectAccess started successfully.
```

Display the status of the object access service.

```
earth.ObjectAccess> server status
ObjectAccess Status on earth_01 : ONLINE
ObjectAccess Status on earth_02 : ONLINE
```

Stop the object access service.

```
earth.ObjectAccess> server stop
ACCESS ObjectAccess SUCCESS V-288-0 ObjectAccess stopped successfully.
```

Disable the object access service.

```
earth.ObjectAccess> server disable
This operation will delete ObjectAccess authentication information, do you
want to continue(y/n): y
100% [********************] Disabling ObjectAccess server.
ACCESS ObjectAccess SUCCESS V-288-0 ObjectAccess server disabled.
```

Display the configuration of the object access service.

```
earth.ObjectAccess> show
Name           Value
=============  =========================
Server Status  Disabled
Admin_URL      http://admin.earth:8144
S3_URL         http://s3.earth:8143
admin_port     8144
s3_port        8143
ssl            no
fs_size        20G
pools          pool1,pool2
fs_blksize     8192
fs_pdirenable  no
fs_encrypt     off
fs_type        mirrored
fs_nmirrors    2
fs_protection  disk
```

List the S3 users.

```
earth.ObjectAccess> account user show
AccessKey            UserName
===================  ================
MzhiNzRlNDQ1MDQ1YjB  user1
```

```
M2UzZmQzMTk1MjU2MDY  nis-user1-group1
ZTMyYTRlYWMwNTE4ZDV  nis-user1-group3
```

List the buckets created by the S3 users.

```
earth.ObjectAccess> bucket show
Bucket Name                Fs Name         Pool         Owner
========================  ==============  ===========  ================
nis-user1-group1bucket13  S3fs1489227391  pool-group1  nis-user1-group1
nis-user1-group1bucket12  S3fs1489224108  pool-s3_2    nis-user1-group1
nis-user1-group1bucket11  S3fs1489224886  pool-s3_2    nis-user1-group1
nis-user1-group1bucket14  S3fs1489227391  pool-group1  nis-user1-group1
nis-user1-group1bucket1   S3fs1489224561  pool-s3_2    nis-user1-group1
moonbucket1               S3fs1489224108  pool-s3_2    user1
nis-user1-group3bucket1   S3fs1489229942  pool-s3_2    nis-user1-group3
```

List the bucket details for the specific bucket.

```
earth.ObjectAccess> bucket show nis-user1-group1bucket13
Bucket Name                Fs Name         Pool         Owner
========================  ==============  ===========  ================
nis-user1-group1bucket13  S3fs1489227391  pool-group1  nis-user1-group1
```

Enable secure socket communication for the object access server.

```
earth.ObjectAccess> set ssl_enabled yes
ACCESS ObjectAccess INFO V-288-0 Set ssl-enabled successful.
ACCESS SSL INFO V-288-0 Note: Please restart the objectaccess server.
```

Set the default pool(s), which are used to create the file system while creating the buckets. This default pool can be overridden by setting up group-specific options by using the `set` command.

```
earth.ObjectAccess> set pools pool1,pool2
ACCESS ObjectAccess INFO V-288-0 Set pools successful. Please make sure the storage␣
↪is provisioned as per the requirements of the layout.
```

Set the default file system size that should be used while creating the file system for a given bucket. This default `fs_size` can be overridden by setting up group-specific options by using the `set` command.

```
earth.ObjectAccess> set fs_size 10G
ACCESS ObjectAccess INFO V-288-0 Set operation successful.
```

Set the group's pool mapping. In the following example, all the buckets of nis-group1 get created on pool-group1.

```
earth.ObjectAccess> group set pool nis-group1 pool-group1
ACCESS ObjectAccess INFO V-288-0 Set pool successful.
```

Set the group's file system type.

```
earth.ObjectAccess> group set fs_type simple nis-group1 blksize=1024 pdir_enable=no␣
↪encrypt=off
ACCESS ObjectAccess INFO V-288-0 Set pool successful.
```

Set the group's file system size.

```
earth.ObjectAccess> group set fs_size nis-group1 5G
ACCESS ObjectAccess INFO V-288-0 Set fs_size successful.
```

Set the group's file system sharing.

```
earth.ObjectAccess> group set fs_sharing nis-group1 yes
ACCESS ObjectAccess INFO V-288-0 Set fs_sharing successful.
```

Unset the group's pool mapping.

```
earth.ObjectAccess> group unset pool nis-group1 pool-group1
ACCESS ObjectAccess INFO V-288-0 group unset pool Successful.
```

Unset the group's fs_sharing.

```
earth.ObjectAccess> group unset fs_sharing nis-group1
ACCESS ObjectAccess INFO V-288-0 group unset fs_sharing Successful.
```

Unset the group's fs_size.

```
earth.ObjectAccess> group unset fs_size nis-group1
ACCESS ObjectAccess INFO V-288-0 group unset fs_size Successful.
```

Show the group's configuration details.

```
earth.ObjectAccess> group show
Group Name   Fs Sharing   Fs Size   Fs Type   Pool
===========  ===========  ========  ========  ===============
nis-group1   yes          5G        simple    pool-group1
```

Map file system path as bucket for a particular user.

```
earth.ObjectAccess> map /vx/fs1/dir1 s3user1
ACCESS ObjectAccess SUCCESS V-288-0 Successfully mapped bucket dir1 to s3user1.
```

11.1.5 SEE ALSO

account(1), group(1), bucket(1), server(1), show(1), set(1), unset(1), map(1)

11.2 account

11.2.1 SYNOPSIS

`account` user show

11.2.2 DESCRIPTION

The ObjectAccess `account` command is used to list the S3 users.

11.2.3 OPTIONS

`account` user show List the S3 users.

11.2.4 EXAMPLES

List the S3 users.

```
earth.ObjectAccess> account user show
AccessKey            UserName
===================  ================
MzhiNzRlNDQ1MDQ1YjB  user1
M2UzZmQzMTk1MjU2MDY  nis-user1-group1
ZTMyYTRlYWMwNTE4ZDV  nis-user1-group3
```

11.2.5 SEE ALSO

objectaccess(1)

11.3 bucket

11.3.1 SYNOPSIS

`bucket` show [bucket_name]

11.3.2 DESCRIPTION

The ObjectAccess `bucket show` command is used to list the buckets created by the S3 users.

11.3.3 OPTIONS

`bucket` [show] List the buckets created by the S3 users.

11.3.4 EXAMPLES

List the buckets created by the S3 users.

```
earth.ObjectAccess> bucket show
Bucket Name                FileSystem         Pool(s)      Owner
========================  ==============  ===========  ================
nis-user1-group1bucket13  S3fs1489227391  pool-group1  nis-user1-group1
nis-user1-group1bucket12  S3fs1489224108  pool-s3_2    nis-user1-group1
nis-user1-group1bucket11  S3fs1489224886  pool-s3_2    nis-user1-group1
nis-user1-group1bucket14  S3fs1489227391  pool-group1  nis-user1-group1
nis-user1-group1bucket1   S3fs1489224561  pool-s3_2    nis-user1-group1
moonbucket1               S3fs1489224108  pool-s3_2    user1
nis-user1-group3bucket1   S3fs1489229942  pool-s3_2    nis-user1-group3
scale-out-bucket1         S3fs1489230130  pool-group2  nis-user1-group4

earth.ObjectAccess> bucket show nis-user1-group1bucket13
Bucket Name                FileSystem         Pool(s)      Owner
========================  ==============  ===========  ================
nis-user1-group1bucket13  S3fs1489227391  pool-group1  nis-user1-group1
```

11.3.5 SEE ALSO

objectaccess(1)

11.4 group

11.4.1 SYNOPSIS

`group` set [fs_sharing|fs_size|fs_type|pools] *group_name value*

`group` show [group_name]

`group` unset *group_name* [fs_sharing|fs_size|fs_type|pool]

11.4.2 DESCRIPTION

The ObjectAccess `group` commands allow an administrator to override the default file system configuration parameters. This gives fine control over how the object access server creates file systems and buckets for particular user groups. It is recommended to create new groups using the authentication server (NIS/LDAP/AD) and use them for this group configuration.

11.4.3 OPTIONS

group set [fs_sharing|fs_size|fs_type|pools] *group_name value* The `fs_sharing` option determines whether to create a new file system or to use an existing file system for creating the bucket. The `fs_size` and `fs_type` option allow you to set size and type of the file system for the group if any configuration parameters are not set. The `pools` option lets you choose the pool on which the bucket has to be created. If any configuration parameter is not set then it will pick from default configuration.

group show [group_name] Lists group specific configurations for all the groups. If group name is specificed, it will give all configuration details for that group.

group unset *group_name* [fs_sharing|fs_size|fs_type|pool] Removes the option set for the specific group.

11.4.4 EXAMPLES

Set the group's pool mapping. In the following example, all the buckets of nis-group1 get created on pool-group1.

```
earth.ObjectAccess> group set pool nis-group1 pool-group1
ACCESS ObjectAccess INFO V-288-0 Set pool successful.
```

Set the group's file system type.

```
earth.ObjectAccess> group set fs_type simple nis-group1 blksize=1024 pdir_enable=no␣
↪encrypt=off
ACCESS ObjectAccess INFO V-288-0 Set pool successful.
```

Set the group's file system size.

```
earth.ObjectAccess> group set fs_size nis-group1 5G
ACCESS ObjectAccess INFO V-288-0 Set fs_size successful.
```

Set the group's file system sharing.

```
earth.ObjectAccess> group set fs_sharing nis-group1 yes
ACCESS ObjectAccess INFO V-288-0 Set fs_sharing successful.
```

Unset the group's pool mapping.

```
earth.ObjectAccess> group unset nis-group1 pool pool-group1
ACCESS ObjectAccess INFO V-288-0 group unset pool Successful.
```

Unset the group's fs_sharing.

```
earth.ObjectAccess> group unset nis-group1 fs_sharing
ACCESS ObjectAccess INFO V-288-0 group unset fs_sharing Successful.
```

Unset the group's fs_type.

```
earth.ObjectAccess> group unset nis-group1 fs_type
ACCESS ObjectAccess INFO V-288-0 group unset fs_type Successful.
```

Unset the group's fs_size.

```
earth.ObjectAccess> group unset nis-group1 fs_size
ACCESS ObjectAccess INFO V-288-0 group unset fs_size Successful.
```

Show the group's configuration details.

```
moon.ObjectAccess> group show
Group Name   Fs Sharing   Fs Size   Fs Type   Pool(s)
===========  ===========  ========  ========  ===========
nis-group1   yes          5G        simple    pool-group1

moon.ObjectAccess> group show nis-group1
Name           Value
=============  ===========
Group          nis-group1
poollist       pool-group1
fs_sharing     yes
fs_size        5G
fs_type        simple
fs_blksize     1024
fs_pdirenable  no
fs_encrypt     off
fs_type        simple
```

11.4.5 SEE ALSO

objectaccess(1)

11.5 map

11.5.1 SYNOPSIS

`map` filesystem_path username

11.5.2 DESCRIPTION

Map the specific path as an S3 bucket for a particular user. The file system path can be either a directory inside a normal file system or an NFS exported path. If the specified path is not present, then the map command creates the directory and sets the default S3 bucket permissions. The ownership of the bucket is set to the user and it's group. The file system path should not be a directory inside the exported NFS share or the existing bucket. The bucket name is the same as the directory name. Admin can check the path mapping with the bucket show command.

11.5.3 OPTIONS

`map` filesystem_path username

Map filesystem_path as a S3 bucket for user username. Path can be any directory path inside the file system or nfs exported path. User can be any authenticated user from AD/LDAP/NIS.

11.5.4 EXAMPLES

Map the file system path in which directory is not present as an S3 bucket.

```
vmdellr> objectaccess map /vx/simple_fs/dir2 domain1/ad-user2
ACCESS ObjectAccess SUCCESS V-288-0 Successfully mapped bucket dir2 to domain1/ad-
↪user2.
```

Map the file system path in which directory is already present as an S3 bucket.

```
vmdellr> objectaccess map /vx/simple_fs/dir6 user2
This operation will change the ownership of directory '/vx/simple_fs/dir6' to user
↪'user2' and will affect existing permissions. Do you want to continue (yes/no)? yes
ACCESS ObjectAccess SUCCESS V-288-0 Successfully mapped bucket dir6 to user2.
```

Map the nfs share path as an S3 bucket.

```
vmdellr> objectaccess map /vx/simple_fs/dir4 user2
This operation will change the ownership of directory '/vx/simple_fs/dir4' to user
↪'user2' and will affect existing permissions. Do you want to continue (yes/no)? yes
ACCESS ObjectAccess SUCCESS V-288-0 Successfully mapped bucket dir4 to user2.
```

11.5.5 SEE ALSO

objectaccess(1)

11.6 server

11.6.1 SYNOPSIS

`server` [enable|start|stop|status|disable]

11.6.2 DESCRIPTION

The ObjectAccess `server` commands are used to start, stop, or check the status of the object access server.

11.6.3 OPTIONS

`server` [enable|start|stop|status|disable] Start, stop, or check the status of the object access service. The server must be enabled before starting. Disabling the server deletes the object access configuration and user information database. All S3 buckets must be deleted before disabling the object access server.

11.6.4 EXAMPLES

Enable the object access service.

```
earth.ObjectAccess> server enable
100% [********************] Enabling ObjectAccess server.
ACCESS ObjectAccess SUCCESS V-288-0 ObjectAccess server enabled.
```

Start the object access service.

```
earth.ObjectAccess> server start
ACCESS ObjectAccess SUCCESS V-288-0 ObjectAccess started successfully.
```

Display the status of the object access service.

```
earth.ObjectAccess> server status
ObjectAccess Status on earth_01 : ONLINE
ObjectAccess Status on earth_02 : ONLINE
```

Stop the object access service.

```
earth.ObjectAccess> server stop
ACCESS ObjectAccess SUCCESS V-288-0 ObjectAccess stopped successfully.
```

Disable the object access service.

```
earth.ObjectAccess> server disable
This operation will delete ObjectAccess authentication information, do you
want to continue(y/n): y
100% [********************] Disabling ObjectAccess server.
ACCESS ObjectAccess SUCCESS V-288-0 ObjectAccess server disabled.
```

11.6.5 SEE ALSO

objectaccess(1)

11.7 set

11.7.1 SYNOPSIS

`set` [ssl_enabled|pools|fs_size|fs_type]

11.7.2 DESCRIPTION

The ObjectAccess commands are used to set object access configurable and default file system options.

11.7.3 OPTIONS

`set` [ssl_enabled|pools|fs_size|fs_type] Set object access configurable and default file system options. The object access server creates a new file system for every new bucket. Use the set command to create a new file system of a different configuration. The `ssl_enabled` option enables HTTPS access. The `pools` option lets you choose the pool on which the bucket has to be created. Use the `fs_size` option to set the default file system size for the buckets. Use the `fs_type` option to choose the type of the file system that is created as part of bucket creation.

11.7.4 EXAMPLES

Set `ssl_enabled` to `yes` to have secure communication between the client and the server.

```
earth.ObjectAccess> set ssl_enabled yes
ACCESS ObjectAccess INFO V-288-0 Set ssl-enabled successful.
ACCESS SSL INFO V-288-0 Note: Please restart the objectaccess server.
```

Set `ssl_enabled` to `no` if you do not want secure communication.

```
earth.ObjectAccess> set ssl_enabled no
ACCESS ObjectAccess INFO V-288-0 Set ssl-enabled successful.
ACCESS SSL INFO V-288-0 Note: Please restart the objectaccess server.
```

Set the default pool(s), which are used to create the file system while creating the buckets. This default pool can be overridden by setting up group-specific options by using the `set` command.

```
earth.ObjectAccess> set pools pool1
ACCESS ObjectAccess INFO V-288-0 Set pools successful. Please make sure the
storage is provisioned as per the requirements of the layout.
```

Set the default `fs_type` to a mirrored layout with 2 mirrors.

```
earth.ObjectAccess> set fs_type mirrored 2 protection=pool blksize=1024 pdir_
↪enable=no encrypt=off
ACCESS ObjectAccess INFO V-288-0 Set fs_type mirrored successful.
```

Set the default `fs_type` to scale-out fs simple layout.

```
    earth.ObjectAccess> set fs_type largefs simple
    ACCESS ObjectAccess INFO V-288-0 Set fs_type successful. Please make sure the
storage is provisioned as per the requirements of the layout.
```

Set the default `fs_type` to scale-out fs erasure coded layout with 4 data columns, 2 parity and stirpe unit of 16k. This configuration for erasure-coded filesystem will be used for creating buckets.

```
    earth.ObjectAccess> set fs_type largefs ecoded 4 2 16k stripe_aligned=yes stripe_
↪tag=disk rotating_parity=yes
    ACCESS ObjectAccess INFO V-288-0 Set fs_type successful. Make sure the
storage is provisioned as per the requirements of the layout.
```

Set the default file system size that should be used while creating the file system for a given bucket. This default `fs_size` can be overridden by setting up group-specific options by using the `set` command.

```
earth.ObjectAccess> set fs_size 10G
ACCESS ObjectAccess INFO V-288-0 Set operation successful.
```

11.7.5 SEE ALSO

objectaccess(1), unset(1)

11.8 show

11.8.1 SYNOPSIS

`show`

11.8.2 DESCRIPTION

The ObjectAccess `show` command is used to list all the object access service configurable options.

11.8.3 OPTIONS

`show` List all the object access configurable options.

11.8.4 EXAMPLES

Display the configuration of the object access service.

```
earth.ObjectAccess> show
Name           Value
=============  =========================
Server Status  Disabled
Admin_URL      http://admin.earth:8144
S3_URL         http://s3.earth:8143
admin_port     8144
s3_port        8143
ssl            no
fs_size        20G
pools          pool1,pool2
fs_blksize     8192
fs_pdirenable  no
fs_encrypt     off
fs_type        mirrored
fs_nmirrors    2
fs_protection  disk
```

11.8.5 SEE ALSO

objectaccess(1), set(1), unset(1)

11.9 unset

11.9.1 SYNOPSIS

`unset` pools pool1[,pool2,...]

11.9.2 DESCRIPTION

The ObjectAccess `unset` command is used to unset the configuration of the object access service.

11.9.3 OPTIONS

`unset` pools pool1[,pool2,...] Remove pools from the default pool list. If any bucket is present in that pool, then the unset command fails for that pool. Removing all the pools is not allowed as the object access service requires at least one default pool.

11.9.4 EXAMPLES

Unset the pool mapping.

```
earth.ObjectAccess> unset pools pool-group1
ACCESS ObjectAccess INFO V-288-0 Unset pools successful.
```

11.9.5 SEE ALSO

objectaccess(1), set(1)

12 OpenStack Commands

12.1 openstack

12.1.1 SYNOPSIS

cinder `configure` *export_dir*

cinder `service start`

cinder `service status`

cinder `share add` *export_dir world|client*

cinder `share delete` *export_dir world|client*

cinder `share show`

manila `configure` *pool_name*

manila `resource export` *pool_name*

manila `resource delete` *pool_name*

manila `resource list`

12.1.2 DESCRIPTION

The `openstack` commands configure Veritas Access to work with OpenStack Cinder or Manila. The OpenStack `cinder` share commands add, delete, or display the NFS share. OpenStack `cinder configure` command displays all the configuration changes that need to be done on the Cinder node to access the Veritas Access storage as a volume backend from Cinder. OpenStack `manila resource` command exports/unexports any existing storage pools of Veritas Access to OpenStack Manila.

12.1.3 OPTIONS

export_dir Path of the directory that needs to be exported to the client. The directory path should start with /vx and only 'a-zA-Z0-9_/@+=.:-' characters are allowed in *export_dir*.

client Export the directory with the specified options. Clients may be specified in the following ways:

1. Single host

   Specify a host either by an abbreviated name recognized by the resolver, the fully qualified domain name, or an IP address.

2. Netgroups

   Netgroups may be given as *@group*. Only the host part of each netgroup member is considered when checking for membership.

3. IP networks

   You can also simultaneously export directories to all hosts on an IP (sub-) network. This is done by specifying an IP address and netmask pair as *address/netmask* where the netmask can be specified as a contiguous mask length. IPv4 or IPv6 addresses can be used.

world With this option, the specified file system will be accessed or mounted by any client. To re-export new options to an existing share, the new options will be updated after the command is run.

pool_name Existing storage pool that has been exported to Manila.

`cinder configure` ***export_dir*** Display all the configuration changes that need to be done on the OpenStack Controller node. After making these modifications on the OpenStack Controller node, you can use the Veritas Access storage as a volume backend.

`cinder service start` Start the OpenStack Cinder service on Veritas Access.

`cinder service status` Display the status of the OpenStack Cinder service on Veritas Access.

`cinder share add` ***export_dir client*** Share and export the file system *export-dir*. After issuing this command, OpenStack Cinder will be able to mount the exported file system using NFS.

`cinder share delete` ***export_dir client*** Delete (or unshare) the exported file system. Share and export /vx/fs1 using NFS to all the clients.

cinder ***export_dir client*** Delete (or unshare) the exported file system. Share and export /vx/fs1 using NFS to all the clients.

`manila configure` ***pool_name*** Display all the configuration changes that need to be done on the OpenStack Controller node, given an exported pool.

`manila resource export` ***pool_name*** Export existing pool pool_name to the Veritas Access OpenStack Manila driver.

`manila resource delete` ***pool_name*** Unexport the exported pool.

`manila resource list` List all the Veritas Access resources created or exported by or to the Manila driver.

12.1.4 EXAMPLES

```
OPENSTACK> configure /vx/fs1
```

To create a new volume backend ACCESS_HDD in OpenStack Cinder

Make the following changes on the OpenStack controller node and restart the Cinder driver.

Add the following configuration entries in the **/etc/cinder/cinder.conf** file:

```
In the [DEFAULT] section:
#####
enabled_backends=access-1
#####
```

At the end of all the sections:

```
#####
[access-1]
volume_driver=cinder.volume.drivers.veritas_cnfs.VeritasCNFSDriver
volume_backend_name=ACCESS_HDD
nfs_shares_config=/etc/cinder/access_share_hdd
nfs_mount_point_base=/cinder/cnfs/cnfs_sata_hdd
nfs_sparsed_volumes=True
nfs_disk_util=df
nfs_mount_options=nfsvers=3
#####
```

Append the following in the **/etc/cinder/access_share_hdd** file:

```
#####
vip:/vx/fs1
#####
```

Use one of the following virtual IPs for `vip`:

```
192.1.1.190
192.1.1.191
192.1.1.192
192.1.1.193


OPENSTACK> cinder service start
..Success.

OPENSTACK> cinder service status
NFS Status on access_01 : ONLINE
NFS Status on access_02 : ONLINE

OPENSTACK> cinder share add /vx/fs1 world
Exporting /vx/fs1 with options rw,no_root_squash
Success.

OPENSTACK> cinder share delete /vx/fs1 world
Removing export path \*:/vx/fs1
Success.

OPENSTACK> cinder share add /vx/fs1 world
Exporting /vx/fs1 with options rw,no_root_squash
Success.

OPENSTACK> manila configure pool1
```

To create a new share backend va-share1 in OpenStack Manila

Make the following changes on the OpenStack controller node and restart the Manila driver. Add the following configuration entries in the **/etc/manila/manila.conf** file:

In the [DEFAULT] section:

```
#####
enabled_share_backends=va-share1
#####
```

At the end of all the sections:

```
#####
[va-share1]
share_driver= manila.share.drivers.veritas.veritas_isa.ACCESSShareDriver
driver_handles_share_servers = False
share_backend_name = va-share1
va_server_ip = <va_console_ip>
va_port = 14161
va_fstype = simple
va_user = <master_user>
va_pwd = <master_user_password>
va_pool = pool1
#####

OPENSTACK> manila resource export pool1
ACCESS Manila SUCCESS V-288-0 Pool exported to Manila.

OPENSTACK> manila resource delete pool1
ACCESS Manila SUCCESS V-288-0 Pool removed from exported list.

OPENSTACK> manila resource list
Pools exported to Manila:
  pool1

FS created by Manila:
  0B54F556-ACE40746
  F4793495-227F6C9B

FS snapshots created by Manila:
  701C1975-AC635AAF

NFS shares exported by Manila:
  /vx/0B54F556-ACE40746
  /vx/0B54F556-ACE40746
```

12.1.5 SEE ALSO

cinder(1), manila(1)

12.2 cinder

12.2.1 SYNOPSIS

`configure` *export_dir*

`service start`

cinder `service status`

cinder `share add` *export_dir world|client*

cinder `share delete` *export_dir world|client*

cinder `share show`

12.2.2 DESCRIPTION

The cinder `configure` command displays all the configuration changes that need be done on the OpenStack Cinder node to access Veritas Access storage as a volume backend from OpenStack Cinder. Only after making these modifications on the OpenStack Cinder node, Cinder can use the Veritas Access share as a volume backend.

The cinder `service start` command starts the NFS service. NFS service has to be started as OpenStack service relies on it. The cinder `serivce status` command displays the status of the OpenStack service. If the NFS service is up, the OpenStack service is up.

The cinder `service share` commands add, delete, or display the NFS share. The OpenStack Cinder node can NFS mount a particular share only after issuing the add command.

Note: The Veritas Access server includes the Veritas Access OpenStack Cinder driver, which is a Python script.

The OpenStack Cinder driver is located at **/opt/VRTSnas/scripts/OpenStack/veritas_cnfs.py** on the Veritas Access server. If you are using the Python 2.6 release, copy the **veritas_cnfs.py** file to **/usr/lib/python2.6/site-packages/cinder/volume/drivers/veritas_cnfs.py** on the OpenStack controller node. If you are using the Python 2.7 release, copy the **veritas_cnfs.py** file to **/usr/lib/python2.7/site-packages/cinder/volume/drivers/veritas_cnfs.py** on the OpenStack controller node.

12.2.3 OPTIONS

export_dir Path of the directory that needs to be exported to the client. The directory path should start with /vx and only 'a-zA-Z0-9_/@+=.:-' characters are allowed in *export_dir*.

client Export the directory with the specified options. Clients may be specified in the following ways:

1. Single host

   Specify a host either by an abbreviated name recognized by the resolver, the fully qualified domain name, or an IP address.

2. Netgroups

   Netgroups may be given as @*group*. Only the host part of each netgroup member is considered when checking for membership.

3. IP networks

   You can also simultaneously export directories to all hosts on an IP (sub-) network. This is done by specifying an IP address and netmask pair as *address/netmask* where the netmask can be specified as a contiguous mask length. IPv4 or IPv6 addresses can be used.

world With this option, the specified file system will be accessed or mounted by any client.

To re-export new options to an existing share, the new options will be updated after the command is run.

`cinder configure` ***export_dir*** Display all the configuration changes that need to be done on the OpenStack Cinder node. Only after these modifications on the OpenStack Cinder node, OpenStack Cinder can use the Veritas Access storage as a volume backend.

`cinder service start` Start the cinder service on Veritas Access.

`cinder service status` Display the status of cinder service on Veritas Access.

`cinder share add` ***export_dir client*** Share and export the file system *export-dir* to *client*. Here *client* is OpenStack controller node. After issuing this command, OpenStack Cinder will be able to mount the exported file system using NFS. By default the file system *export-dir* is exported with no_root_squash option.

`cinder share add` ***export_dir*** **world** Share and export the file system *export-dir* to all the clients. After issuing this command, any node will be able to mount the exported file system using NFS.

`cinder share delete` ***export_dir client*** Delete (or unshare) the exported file system. Share and export /vx/fs1 using NFS to all the clients.

`cinder share delete` ***export_dir*** **world** Unexport the exported file system to all clients. If a file system is exported specifically to some clients, those clients would still access the file system.

12.2.4 EXAMPLES

```
OPENSTACK> cinder configure /vx/fs1
```

To create a new volume backend ACCESS_HDD in Cinder

Make the following changes on the OpenStack controller node and restart the Cinder driver.

Add the following configuration entries in the **/etc/cinder/cinder.conf** file: In the [DEFAULT] section:

```
#####
enabled_backends=access-1
#####
```

At the end of all the sections:

```
#####
[access-1]
volume_driver=cinder.volume.drivers.veritas_cnfs.VeritasCNFSDriver
volume_backend_name=ACCESS_HDD
nfs_shares_config=/etc/cinder/access_share_hdd
nfs_mount_point_base=/cinder/cnfs/cnfs_sata_hdd
nfs_sparsed_volumes=True
nfs_disk_util=df
nfs_mount_options=nfsvers=3
#####
```

Append the following in the **/etc/cinder/access_share_hdd** file:

```
#####
vip:/vx/fs1
#####
```

Use one of the following virtual IPs for `vip`.

```
192.1.1.190
192.1.1.191
192.1.1.192
192.1.1.193


OPENSTACK> cinder service start
..Success.

OPENSTACK> cinder service status
NFS Status on access_01 : ONLINE
NFS Status on access_02 : ONLINE

OPENSTACK> cinder share add /vx/fs1 world
Exporting /vx/fs1 with options rw,no_root_squash
Success.

OPENSTACK> cinder share delete /vx/fs1 world
Removing export path *:/vx/fs1
Success.

OPENSTACK> cinder share add /vx/fs1 world
Exporting /vx/fs1 with options rw,no_root_squash
Success.

OPENSTACK> cinder share show
/vx/fs1                                          * (rw,no_root_squash)
```

12.3 manila

12.3.1 SYNOPSIS

`manila resource export` *pool_name*

`manila resource delete` *pool_name*

`manila resource list`

`manila configure` *pool_name*

12.3.2 DESCRIPTION

These commands configure Veritas Access to work with OpenStack Manila. manila `configure` command displays all the configuration changes that need be done on the OpenStack controller node for the Manila service to communicate with Veritas Accessfor share creation and management.

The Manila `resource export` command exports any existing storage pools of Veritas Access to OpenStack Manila. The manila `resource delete` command unexports the exported pool provided to OpenStack. Veritas Access keeps record of all the shares, snapshots and NFS exports created by OpenStack manila driver, `resource list` displays all the resources created by the Manila driver.

Note: The Veritas Access server includes the Veritas Access OpenStack Manila driver, which is a Python script. The OpenStack Manila driver is located at **/opt/VRTSnas/scripts/OpenStack/veritas/veritas_isa.py** on the Veritas Access server. If you are using the Python 2.6 release, copy the **veritas** folder to **/usr/lib/python2.6/site-packages/manila/share/drivers/** on the OpenStack controller node. If you are using the Python 2.7 release, copy the **veritas** folder to **/usr/lib/python2.7/site-packages/manila/share/drivers/** on the OpenStack controller node.

`manila configure` ***pool_name*** Display all the configuration changes that need to be done on the OpenStack Controller node.

`manila resource export` ***pool_name*** Export existing pool `pool_name` to the Veritas Access OpenStack Manila driver.

`manila resource delete` ***pool_name*** Unexport the exported pool.

`manila resource list` List all the Veritas Access resources created or exported by or to the Manila driver.

12.3.3 EXAMPLES

Display all the configuration changes that need to be done on the OpenStack controller node to make `pool1` as the share backend.

```
OPENSTACK> manila configure pool1
```

To create a new share backend va-share1 in OpenStack Manila

Make the following changes on OpenStack controller node and restart the Manila driver Add the following configuration entries in the **/etc/manila/manila.conf** file:

```
In the [DEFAULT] section:
#####
enabled_share_backends=va-share1
#####
```

At the end of all the sections:

```
#####
[va-share1]
share_driver= manila.share.drivers.veritas.veritas_isa.ACCESSShareDriver
driver_handles_share_servers = False
share_backend_name = va-share1
va_server_ip = <va_console_ip>
va_port = 14161
va_fstype = simple
va_user = <master_user>
va_pwd = <master_user_password>
va_pool = pool1
#####
```

```
OPENSTACK> manila resource export pool1
ACCESS REST SUCCESS V-288-1397 Export pool pool1 to REST server.

OPENSTACK> manila resource delete pool1
ACCESS REST SUCCESS V-288-1397 Removing exported pool pool1 from REST server.

OPENSTACK> manila resource list
Pools exported to Manila:
  pool1

FS created by Manila:
  0B54F556-ACE40746
  F4793495-227F6C9B

FS snapshots created by Manila:
  701C1975-AC635AAF

NFS shares exported by Manila:
  /vx/0B54F556-ACE40746
  /vx/0B54F556-ACE40746
```

13

OpenDedup Commands

13.1 opendedup

13.1.1 SYNOPSIS

`opendedup volume create volume-name volume-size cloud-access-key cloud-bucket-name [cache-volsize] [cache-voltype]`

`opendedup volume list` [volume-name]

`opendedup volume online volume-name`

`opendedup volume offline volume-name`

`opendedup volume delete volume-name`

13.1.2 DESCRIPTION

The `opendedup volume` commands are used to perform sdfs volume-related operations.

13.1.3 OPTIONS

`opendedup volume create volume-name volume-size cloud-access-key`

`cloud-bucket-name [cache-volsize] [cache-voltype]` Create a sdfs volume with the specified name, cloud access key, cloud bucket name, and the size for the specified file system. Internally, it creates the "odd_cache_fs" file system which is required for OpenDedup. The default size is of this file system is 24 GB but it can be set to any size based on performance and storage requirements by the user. Accordingly, default layout of this file system is "mirrored", but can be set to "simple" as well based on requirements by the user. Every time an OpenDedup volume is created, the size of this file system increases. But the deletion of this file system does not reclaim the storage. For reclaiming the storage, the "odd_cache_fs" file system needs to be shrunk when the I/O load on OpenDedup is low.

`opendedup volume list` [volume-name] List the details of all the volume(s) present in the cluster for a specified volume.

`opendedup volume delete volume-name` Delete a specific volume.

opendedup volume online volume-name Online a specific volume.

opendedup volume offline volume-name Offline a specific volume.

13.1.4 EXAMPLES

```
VA.ODD> volume create TEST_ANTONY_1 100MB MzNjZDMwZTlkNjVkMGM mytfs 100MB
    ACCESS odd SUCCESS V-288-0 TEST_ANTONY_1 has been created successfully.

    VA.ODD> volume list
    Volume   State    Port   S3 Bucket              Node IP
    =======  =======  =====  ==========             ===========
    JAGVOL   Online   6442   MzNjZDMwZTlkNjVkMGM    10.10.10.10
    odpool   Offline  6443   acNjZdybNlkNjVkKGb     20.20.20.20


    VA.ODD> volume list pool0
    Files : 0
    Volume Capacity : 2 GB
    Volume Current Logical Size : 0 B
    Volume Max Percentage Full : 95.0%
    Volume Duplicate Data Written : 0 B
    Unique Blocks Stored: 0 B
    Unique Blocks Stored after Compression : 0 B
    Cluster Block Copies : 2
    Volume Virtual Dedup Rate (Unique Blocks Stored/Current Size) : 0%
    Volume Actual Storage Savings (Compressed Unique Blocks Stored/Current Size): 0%
    Compression Rate: 0%


VA.ODD> volume online pool0
    ACCESS odd SUCCESS V-288-0 sdfs pool0 is online successfully.

    VA.ODD> volume offline pool0
    ACCESS odd SUCCESS V-288-0 sdfs is offline successfully.

    VA.ODD> volume delete pool0
    ACCESS odd SUCCESS V-288-0 The Volume is deleted
```

13.1.5 SEE ALSO

opendedup(1), opendedup_volume(1)

14 Report Commands

14.1 report

14.1.1 SYNOPSIS

`email show` [*group*]

`email add` email-address|filter|group|severity *group*

[command-specific-options]

`email del` email-address|filter|group|severity *group*

[command-specific-options]

`email get`

`email set` [*email-server*] [*email-user*]

`event get dup-frequency`

`event get dup-number`

`event set dup-frequency` *number*

`event set dup-number` *number*

`showevents` [*number_of_events*]

`syslog add` *syslog-server-ipaddr*

`syslog delete` *syslog-server-ipaddr*

`syslog set` filter|severity *value*

`syslog get` filter|severity

`syslog show`

`snmp add` *snmp-mgmtserver-ipaddr*

`snmp delete` *snmp-mgmtserver-ipaddr*

`snmp set` filter|severity *value*

`snmp get` filter|severity

`snmp show`

`snmp exportmib` *url*

`exportevents` *url*

14.1.2 DESCRIPTION

The `report` commands configure the notifications of events. The notifications are based on email notifications, SNMP traps, and passing a syslog of the events to the external syslog server. The commands also support showing the recent events. These events are centrally located.

14.1.3 OPTIONS

`email show` **[*group*]** Show email groups or show an email group's details.

`email add` **email-address|filter|group|severity** ***group*** **[command-specific-options]** Add an email group, or add an email address, filter, or severity to an email group.

`email del` **email-address|filter|group|severity** ***group*** **[command-specific-options]** Delete an email group, or delete an email-address, filter, or severity from an email group.

`email get` Get the details of the configured external email server.

`email set` **[*email-server*] [*email-user*]** Set the details of an external email server. If email-server and email-user are not specified, previously configured settings are deleted.

`event get dup-frequency` View the time interval in which only one event (of duplicate events) is sent for notifications.

`event get dup-number` View the number of duplicate events to ignore during notifications.

`event set dup-frequency` ***number*** Set the time interval in which only one event (of duplicate events) is sent for notifications.

`event set dup-number` ***number*** Set the number of duplicate events to ignore during notifications.

`showevents` **[*number_of_events*]** Display the latest events.

`syslog add` ***syslog-server-ipaddr*** Add an external syslog server.

`syslog delete` ***syslog-server-ipaddr*** Delete a syslog server from the configured syslog servers.

`syslog set filter` ***value*** Set the filter for the syslog notifications. Events matching with given filter are sent for notifications.

`syslog set severity` ***value*** Set the severity for syslog notifications. Events having the same or higher severity are sent for notifications.

`syslog get` **filter|severity** Get the values of the configured settings.

`syslog show` Display the current list of external syslog servers.

`snmp add` ***snmp-mgmtserver-ipaddr*** Add an SNMP management server.

`snmp delete` ***snmp-mgmtserver-ipaddr*** Delete an already configured SNMP management server.

`snmp set filter` ***value*** Set the filter for SNMP notifications. Events matching with given filter are sent for notifications.

`snmp set severity` ***value*** Set severity for SNMP notifications. Events having the same or higher severity are sent for notifications.

snmp get filter|severity Get the values of the configured settings.

snmp show Display the current list of SNMP management servers.

snmp exportmib *url* Upload the SNMP MIB file to a given URL. The URL supports FTP and SCP. If the URL specifies the remote directory, the default file name is access_mib.txt.

exportevents *url* Export the events in syslog format to a given URL. The URL supports FTP and SCP. If the URL specifies a remote directory, the default file name is access_event.log when exporting the events.

14.1.4 EXAMPLES

Add an email group.

```
Report> email add group root
OK Completed
```

Add an email address to the group root.

```
Report> email add email-address root adminuser@localhost
OK Completed
```

Add a filter to an email group.

```
Report> email add filter root storage
OK Completed
```

Add a severity to an email group.

```
Report> email add severity root debug
OK Completed
```

Delete an existing email address from the email group.

```
Report> email del email-address root testuser@localhost
OK Completed
```

Try to delete a non-existent email address from the email group.

```
Report> email del email-address root testuser@localhos
ACCESS email del email-address ERROR V-288-50 Cannot remove email-address, it doesn't␣
↪exist.
```

Set the details of the external email server.

```
Report> email set smtp.veritas.com adminuser
OK Completed
Enter password for user 'adminuser': *********
```

Get the details of the external email server.

```
Report> email get
E-Mail Server: smtp.veritas.com
E-Mail Username: adminuser
E-mail User's Password: ********
OK Completed
```

List the current configured email groups.

```
Report> email show
root
OK Completed
```

Display configuration of email group root.

```
Report> email show root
Group Name: root
Severity of the events: info,debug
Filter of the events: all,storage
Email addresses in the group: adminuser@localhost
OK Completed
```

View the duplicate events frequency.

```
Report> event get dup-frequency
Duplicate events frequency (in seconds): 60 (default)
OK Completed
```

View the number of duplicate events.

```
Report> event get dup-number
Duplicate number of events: 5 (default)
OK Completed
```

Set the duplicate events frequency to 120 seconds.

```
Report> event set dup-frequency 120
OK Completed
```

Set the number of duplicate events to 10.

```
Report> event set dup-number 10
OK Completed
```

View the newly set duplicate events frequency.

```
Report> event get dup-frequency
Duplicate events frequency (in seconds): 120
OK Completed
```

View the newly set number of duplicate events.

```
Report> event get dup-number
Duplicate number of events: 10
OK Completed
```

Display the last 10 events

```
Report> showevents 10
Mar 28 23:11:28 [node1,alert,test] Filesystem accounts1 disabled on node2.
......
OK Completed
```

Add a new syslog server

```
Report> syslog add syslog1.veritas.com
OK Completed
```

Delete a configured syslog server.

```
Report> syslog delete syslog2.veritas.com
OK Completed
```

Show the configured syslog servers.

```
Report> syslog show
Configured syslog servers: syslog1.veritas.com,syslog2.veritas.com
OK Completed
```

Set the syslog events filter.

```
Report> syslog set filter storage
OK Completed
```

Get the configured syslog events filter.

```
Report> syslog get filter
Filter for the events: storage
OK Completed
```

Set the syslog events severity.

```
Report> syslog set severity err
OK Completed
```

Get the configured syslog events' severity.

```
Report> syslog get severity
Severity of the events: err
OK Completed
```

Add a new SNMP management server.

```
Report> snmp add mgmtserv1.veritas.com
OK Completed
```

Delete a configured SNMP management server.

```
Report> snmp delete mgmtserv2.veritas.com
OK Completed
```

Pass invalid value for the SNMP server deletion.

```
Report> snmp delete mgmtserv22.veritas.com
ACCESS snmp delete ERROR V-288-26 Cannot delete SNMP management server, it doesn't␣
↪exist.
```

Show configured SNMP management servers.

```
Report> snmp show
Configured SNMP management servers:
mgmtserv1.veritas.com, mgmtserv2.veritas.com
OK Completed
```

Upload the SNMP MIB file:

```
Report> snmp exportmib scp://admin@server1.veritas.com:/tmp/access_mib.txt
Password: *****
OK Completed
```

Upload the events file in syslog format:

```
Report> exportevents scp://root@server1.veritas.com:/exportevents/event.1
Password: *****
OK Completed
```

Set the SNMP events filter:

```
Report> snmp set filter all
OK Completed
```

Get the configured SNMP events filter:

```
Report> snmp get filter
Filter for the events: all
OK Completed
```

Set the SNMP events' severity:

```
Report> snmp set severity info
OK Completed
```

Get the configured SNMP events' severity:

```
Report> snmp get severity
Severity of the events: info
OK Completed
```

14.1.5 SEE ALSO

email(1), exportevents(1), showevents(1), snmp(1), syslog(1), event(1)

14.2 email

14.2.1 SYNOPSIS

`email show` [*group*]

`email add` email-address|filter|group|severity *group*

[command-specific-options]

`email del` email-address|filter|group|severity *group*

[command-specific-options]

`email get`

`email set` [*email-server*] [*email-user*]

14.2.2 DESCRIPTION

The report `email` commands configure email notifications of events. These commands support adding email groups, adding filters to the group, adding email addresses to the email group, adding event severity to the group, and configuring an external email server for sending the event notification emails.

14.2.3 OPTIONS

group Name of the group to create, or to add email addresses.

email-server IP address or hostname of the external email server. IPv6 address should be enclosed by square brackets [].

email-user Username of the external email server.

`email show` [*group*] Show email groups or show an email group's details.

`email add` email-address|filter|group|severity *group* [command-specific-options] Add an email group, or add an email address, filter, or severity to an email group.

`email del` email-address|filter|group|severity *group* [command-specific-options] Delete an email group, or delete an email-address, filter, or severity from an email group.

`email get` Get the details of the configured external email server.

`email set` [*email-server*] [*email-user*] Set the details of external email server. If email-server and email-user are not specified, previously configured settings will be deleted.

14.2.4 EXAMPLES

Add an email group.

```
Report> email add group root
OK Completed
```

Add an email address to the group root.

```
Report> email add email-address root adminuser@localhost
OK Completed
```

Add a filter to an email group.

```
Report> email add filter root storage
OK Completed
```

Add a severity to an email group.

```
Report> email add severity root debug
OK Completed
```

Delete an existing email address from the email group.

```
Report> email del email-address root testuser@localhost
OK Completed
```

Try to delete a non-existant email address from the email group.

```
Report> email del email-address root testuser@localhos
ACCESS email del email-address ERROR V-288-50 Cannot remove email-address, it doesn't␣
↪exist.
```

Set the details of the external email server.

```
Report> email set smtp.veritas.com adminuser
Enter password for user 'adminuser': *********
```

Get the details of the external email server.

```
Report> email get
E-Mail Server: smtp.veritas.com
E-Mail Username: adminuser
E-mail User's Password: ********
OK Completed
```

List the current configured email groups.

```
Report> email show
root
OK Completed
```

Display configuration of email group root.

```
Report> email show root
Group Name: root
Severity of the events: info,debug
Filter of the events: all,storage
Email addresses in the group: adminuser@localhost
OK Completed
```

14.2.5 SEE ALSO

exportevents(1), showevents(1), snmp(1), syslog(1), event(1)

14.3 event

14.3.1 SYNOPSIS

`event get dup-frequency`

`event get dup-number`

`event get log-scan-frequency`

`event get from-address`

`event set dup-frequency` *number*

`event set dup-number` *number*

`event set log-scan-frequency` *frequency*

`event set from-address` *from-email-address*

14.3.2 DESCRIPTION

The report `event` command configures the settings for event reporting.

14.3.3 OPTIONS

`event get dup-frequency` View the time interval in which only one event (of duplicate events) is sent for notifications.

`event get dup-number` View the number of duplicate events to ignore during notifications.

`event get log-scan-frequency` View the log scan frequency.

`event get from-address` View the from email address when sending email to user. Set the from-address to default *cluster_name@domain_name* if set to empty

`event set dup-frequency` *number* Set the time interval in which only one event (of duplicate events) is sent for notifications.

`event set dup-number` *number* Set the number of duplicate events to ignore during notifications.

`event set log-scan-frequency` *frequency* Set the time interval in which the event notify process scan /var/log/messages.

`event set from-address` *from-email-address* Set the from email address when sending email to user.

14.3.4 EXAMPLES

View the duplicate events frequency.

```
Report> event get dup-frequency
Duplicate events frequency (in seconds): 60 (default)
Command completed successfully
```

View the number of duplicate events.

```
Report> event get dup-number
Duplicate number of events: 5 (default)
Command completed successfully
```

View the log scan frequency.

```
Report> event get log-scan-frequency
Log scan frequency (in seconds): 120 (default)
Command completed successfully
```

View the from email address.

```
Report> event get from-address
Email from address: EXAMPLE@access.smtc.com (default)
Command completed successfully
```

Set the duplicate events frequency to 120 seconds.

```
Report> event set dup-frequency 120
Command completed successfully
```

Set the number of duplicate events to 10.

```
Report> event set dup-number 10
Command completed successfully
```

Set the log scan frequency to 30.

```
Report> event set log-scan-frequency 30
Command completed successfully
```

Set the from email address to testaddr@veritas.com.

```
Report> event set from-address testaddr@veritas.com
Command completed successfully
```

View the newly set duplicate events frequency.

```
Report> event get dup-frequency
Duplicate events frequency (in seconds): 120
Command completed successfully
```

View the newly set number of duplicate events.

```
Report> event get dup-number
Duplicate number of events: 10
Command completed successfully
```

View the newly set log scan frequency.

```
Report> event get log-scan-frequency
Log scan frequency (in seconds): 30
Command completed successfully
```

View the newly set from email address.

```
Report> event get from-address
Email from address: testaddr@veritas.com
Command completed successfully
```

14.3.5 SEE ALSO

email(1), exportevents(1), showevents(1), snmp(1), syslog(1)

14.4 exportevents

14.4.1 SYNOPSIS

`exportevents` *url*

14.4.2 DESCRIPTION

The report `exportevents` command exports events in syslog format to the specified URL.

14.4.3 OPTIONS

url URL location to upload the events file in syslog format.

`exportevents` ***url*** Export the events in syslog format to the specified URL. URL supports FTP and SCP. If the URL specifies the remote directory, the default filename is access_event.log.

14.4.4 EXAMPLES

Upload the events file in syslog format.

```
Report> exportevents scp://root@server1.veritas.com:/exportevents/event.1
Password:
File successfully exported to remote location
scp://root@10.209.133.242:/root/event.1
```

14.4.5 SEE ALSO

email(1), snmp(1), syslog(1), showevents(1), event(1)

14.5 showevents

14.5.1 SYNOPSIS

`showevents` [*number_of_events*]

14.5.2 DESCRIPTION

The report `showevents` command displays the latest events.

14.5.3 OPTIONS

number_of_events Number of events to display. Entering 0 shows all of the events.

`showevents` [*number_of_events*] Display latest events.

14.5.4 EXAMPLES

Display the last 10 events:

```
Report> showevents 10
November 12 17:30:24 [test_01,info,root] Disk list detail completed
November 12 17:30:35 [test_01,info,root] Disk list types completed
November 12 17:30:41 [test_01,info,root] Checked status of IO Fencing on the␣
↪coordinator disks
November 12 17:30:54 [test_01,info,root] Disk list stats completed
November 12 17:32:20 [test_01,info,root] CIFS> share show
November 12 17:35:47 [test_01,alert,master] Could not connect
November 12 17:36:56 [test_01,info,test] Notification daemon started.
November 12 17:38:03 [test_01,alert,master] ERROR
November 12 17:38:17 [test_01,alert,master] Could not connect
November 12 17:38:54 [test_01,info,master] ping completed
......
```

14.5.5 SEE ALSO

email(1), exportevents(1), snmp(1), syslog(1), event(1)

14.6 snmp

14.6.1 SYNOPSIS

`snmp add` *snmp-mgmtserver-ipaddr* [community_string]

`snmp delete` *snmp-mgmtserver-ipaddr*

`snmp set` filter|severity *value*

`snmp get` filter|severity

`snmp show`

`snmp exportmib` *url*

14.6.2 DESCRIPTION

The report `snmp` command configures SNMP management servers and filter for notifications, and severity of notifications. The event notifications matching configured severity and filter are sent as SNMP traps to the servers configured with these commands.

14.6.3 OPTIONS

snmp-mgmtserver-ipaddr Host name or IP address of the SNMP management server.

url URL where to upload the SNMP MIB file.

value Value of the variable to set.

`snmp add` *snmp-mgmtserver-ipaddr* [community_string] Add an SNMP management server. The default community_string is 'public'.

`snmp delete` *snmp-mgmtserver-ipaddr* Delete an already configured SNMP management server.

`snmp set filter` *value* Set filter for snmp notifications. Events matching with given filter will be sent for notifications.

`snmp set severity` *value* Set severity for snmp notifications. Events having same or higher severity will be sent for notifications.

`snmp get` filter|severity Get the values of the configured settings.

`snmp show` Show the current list of SNMP management servers.

`snmp exportmib` *url* Upload the SNMP MIB file to the given URL. The URL supports FTP and SCP. If the URL specifies a remote directory, the default filename is access_mib.txt.

14.6.4 EXAMPLES

Add a new SNMP management server.

```
Report> snmp add mgmtserv1.veritas.com
OK Completed
```

Delete a configured SNMP management server.

```
Report> snmp delete mgmtserv2.veritas.com
OK Completed
```

Pass invalid value for SNMP server deletion

```
Report> snmp delete mgmtserv22.veritas.com
ACCESS snmp delete ERROR V-288-26 Cannot delete SNMP management server, it doesn't␣
↪exist.
```

Show the configured SNMP management servers.

```
Report> snmp show
Configured SNMP management servers:
public@mgmtserv1.veritas.com, public@mgmtserv2.veritas.com
OK Completed
```

Upload the SNMP MIB file:

```
Report> snmp exportmib scp://admin@server1.veritas.com:/tmp/access_mib.txt
Password: *****
OK Completed
```

Set SNMP events filter:

```
Report> snmp set filter all
OK Completed
```

Get configured SNMP events filter:

```
Report> snmp get filter
Filter for the events: all
OK Completed
```

Set SNMP events severity:

```
Report> snmp set severity info
OK Completed
```

Get configured SNMP events severity:

```
Report> snmp get severity
Severity of the events: info
OK Completed
```

14.6.5 SEE ALSO

email(1), exportevents(1), showevents(1), syslog(1), event(1)

14.7 syslog

14.7.1 SYNOPSIS

`syslog add` *syslog-server-ipaddr*

`syslog delete` *syslog-server-ipaddr*

`syslog set` filter|severity *value*

`syslog get` filter|severity

`syslog show`

14.7.2 DESCRIPTION

The report `syslog` command configures external syslog servers, the filter for the events and the severity of the events. The event notifications matching configured severity and filter are logged to those external syslog servers.

14.7.3 OPTIONS

syslog-server-ipaddr Hostname or IP address of the external syslog server.

value Value of the variable to set.

`syslog add` ***syslog-server-ipaddr*** Add an external syslog server.

`syslog delete` ***syslog-server-ipaddr*** Delete a syslog server from the configured syslog servers.

`syslog set filter` ***value*** Set filter for syslog notifications. Events matching with given filter will be sent for notifications.

`syslog set severity` ***value*** Set severity for syslog notifications. Events having same or higher severity will be sent for notifications.

`syslog get` **filter|severity** Get the values of the configured settings.

`syslog show` Show the current list of external syslog servers.

14.7.4 EXAMPLES

Add a new syslog server.

```
Report> syslog add syslog1.veritas.com
```

Delete a configured syslog server.

```
Report> syslog delete syslog2.veritas.com
```

Show configured syslog servers.

```
Report> syslog show
Configured syslog servers:
syslog1.veritas.com,syslog2.veritas.com
```

Set syslog events filter:

```
Report> syslog set filter storage
OK Completed
```

Get configured syslog events filter:

```
Report> syslog get filter
Filter for the events: storage
OK Completed
```

Set syslog events severity:

```
Report> syslog set severity err
OK Completed
```

Get configured syslog events severity:

```
Report> syslog get severity
Severity of the events: err
OK Completed
```

14.7.5 SEE ALSO

email(1), exportevents(1), snmp(1), showevents(1), event(1)

15 Replication Commands

15.1 replication

15.1.1 SYNOPSIS

`episodic config bind` *ip_addr* [*device*] [*netmask*]

`episodic config unbind` *ip_addr*

`episodic config show` [*ip* | *remote_clus*]

`episodic config import_keys` [*URL*]

`episodic config export_keys` [*URL*]

`episodic config del_keys` *console_ip*

`episodic config auth` *console_ip link_name*

`episodic config deauth` *link_name*

`episodic config check` *link_name*

`episodic config certificate generate`

`episodic config certificate import`

`episodic config certificate delete`

`episodic config certificate status`

`episodic service start` [*node_name*]

`episodic service stop`

`episodic service status`

`episodic repunit create` *repunit_name repunit_entry*[,*repunit_entry,...*]

`episodic repunit add_entry` *repunit_name repunit_entry*

`episodic repunit add_pattern_entry` *repunit_name repunit_pattern_entry*

`episodic repunit remove_entry` *repunit_name repunit_entry*

`episodic repunit remove_pattern_entry` *repunit_name repunit_pattern_entry*

`episodic repunit modify_entry` *repunit_name old_repunit_entry new_repunit_entry*

`episodic repunit modify_pattern_entry` *repunit_name old_repunit_pattern_entry new_repunit_pattern_entry*

`episodic repunit destroy` *repunit_name* [*option*]

`episodic repunit show` [*repunit_name*] | [*verbose*]

`episodic exclunit create` *exclunit_name exclunit_entry*[,*exclunit_entry,...*]

`episodic exclunit add_entry` *exclunit_name exclunit_entry*

`episodic exclunit add_pattern_entry` *exclunit_name exclunit_pattern_entry*

`episodic exclunit remove_entry` *exclunit_name exclunit_entry*

`episodic exclunit remove_pattern_entry` *exclunit_name exclunit_pattern_entry*

`episodic exclunit modify_entry` *exclunit_name old_exclunit_entry new_exclunit_entry*

`episodic exclunit modify_pattern_entry` *exclunit_name old_exclunit_pattern_entry new_exclunit_pattern_entry*

`episodic exclunit destroy` *exclunit_name*

`episodic exclunit show` [*exclunit_name*]

`episodic schedule create` *schedule_name minute* [*hour*] [*day_of_the_month*] [*month*] [*day_of_the_week*]

`episodic schedule modify` *schedule_name minute* [*hour*] [*day_of_the_month*] [*month*] [*day_of_the_week*]

`episodic schedule show` [*schedule_name*]

`episodic schedule delete` *schedule_name*

`episodic job create` *job_name src_repunit tgt_repunit link_name schedule_name* [*evpsn*] [*metadata_only*]

`episodic job show` [*job_name*]

`episodic job enable` *job_name*

`episodic job sync` *job_name*

`episodic job status` *job_name*

`episodic job pause` *job_name*

`episodic job resume` *job_name*

`episodic job disable` *job_name*

`episodic job destory` *job_name* [*option*]

`episodic job abort` *job_name*

`episodic job failover` *force job_name* [*link_name*]

`episodic job failback` *force job_name* [*link_name*]

`episodic job exclude` *job_name exclunit_name*

`episodic job modify` *debug job_name on\off*

`episodic job modify` *tunables job_name netconn rwcount*

`continuous enable` *fs_name pool_name link_name*

`continuous disable` *fs_name link_name*

`continuous start` *fs_name*

`continuous stop` *fs_name*

`continuous pause` *fs_name*

`continuous resume` *fs_name*

`continuous status` *fs_name*

`continuous failover` *fs_name*

`continuous failback` *fs_name*

`continuous show`

`continuous service start`

`continuous service stop`

`continuous service status`

15.1.2 DESCRIPTION

These `replication` commands are used to manage the replication service on a cluster.

15.1.3 OPTIONS

`episodic config bind` *ip_addr* [*device*] [*netmask*]

Add a virtual IP address to be used as the replication service virtual IP address and *device* as a network interface for the virtual IP address.

`episodic config unbind` *ip_addr*

Delete the virtual IP address from the replication service.

`episodic config export_keys` *URL*

Copy the replication public key from the local cluster to a specified URL.

`episodic config import_keys` *URL*

Copy the replication public key of the remote cluster from a specified URL to a local cluster.

`episodic config del_keys` *console_ip*

Delete the replication public key of a remote cluster identified by *console_ip* from the local cluster.

`episodic config auth` *console_ip link_name*

Authenticate two clusters mutually to participate in the replication service. The * console_ip * is the console IP address of the remote cluster, and *link_name* is the association name between the local cluster and the remote cluster.

`episodic config deauth` *link_name*

Unauthenticate two clusters identified by the association *link_name* from participating in the replication service.

`episodic config check` *link_name*

Check the status of the association identified by the name *link_name* between the source and the target clusters.

`episodic config certificate generate`

Generate a self-signed certificate.

`episodic config certificate import`

Import trusted certificates from an external entity.

`episodic config certificate delete`

Delete configured certificates.

`episodic config certificate status`

Display certificate information.

`episodic service` start [node_name]

Start the episodic replication service. You can use the optional parameter "node_name" to start the replication service on a specific node of the cluster. Otherwise the service start command will automatically figure out the best suitable node in the cluster and will start the replication service there.

`episodic service` stop

Stop the episodic replication service.

`episodic service` status

Check the status of the episodic replication service.

`episodic repunit create` *repunit_name repunit_entry*[,*repunit_entry,...]*

Create a replication unit with a given set of entries.

`episodic repunit add_entry` *repunit_name repunit_entry*

Add a new entry to an existing replication unit.

`episodic repunit add_pattern_entry` *repunit_name repunit_pattern_entry*

Add a new pattern entry to an existing replication unit.

`episodic repunit remove_entry` *repunit_name repunit_entry*

Remove an entry from an existing replication unit.

`episodic repunit remove_pattern_entry` *repunit_name repunit_pattern_entry*

Remove a pattern entry from an existing replication unit.

`episodic repunit modify_entry` *repunit_name old_repunit_entry new_repunit_entry*

Modify an existing entry in a replication unit.

`episodic repunit modify_pattern_entry` *repunit_name old_repunit_pattern_entry new_repunit_pattern_entry*

Modify an existing pattern entry in a replication unit.

`episodic repunit destroy` fIrepunit_name [*option*]

Destroy a replication unit.

`episodic repunit show` [*repunit_name*] | [*verbose*]

Display replication unit details.

`episodic exclunit create` *exclunit_name exclunit_entry*[,*exclunit_entry,...]*

Create an excluding unit with a given set of entries.

`episodic exclunit add_entry` *exclunit_name exclunit_entry*

Add a new entry to an existing excluding unit.

`episodic exclunit add_pattern_entry` *exclunit_name exclunit_pattern_entry*

Add a new pattern entry to an existing excluding unit.

`episodic exclunit remove_entry` *exclunit_name exclunit_entry*

Remove an entry from an existing excluding unit.

`episodic exclunit remove_pattern_entry` *exclunit_name exclunit_pattern_entry*

Remove a pattern entry from an existing excluding unit.

`episodic exclunit modify_entry` *exclunit_name old_exclunit_entry new_exclunit_entry*

Modify an existing entry in an excluding unit.

`episodic exclunit modify_pattern_entry` *exclunit_name old_exclunit_pattern_entry new_exclunit_pattern_entry*

Modify an existing pattern entry in an excluding unit.

`episodic exclunit destroy` fIexclunit_name

Destroy an excluding unit.

`episodic exclunit show` [*exclunit_name*]

Display excluding unit details.

`episodic schedule create` *schedule_name minute* [*hour*] [*day_of_the_month*] [*month*] [*day_of_the_week*]

Create a schedule.

`episodic schedule modify` *schedule_name minute* [*hour*] [*day_of_the_month*] [*month*] [*day_of_the_week*]

Modify an existing schedule.

`episodic schedule show` *schedule_name*

Show a schedule.

`episodic schedule delete` *schedule_name*

Delete a schedule.

`episodic job create` *job_name src_repunit tgt_repunit link_name schedule_name* [*evpsn*] [*metadata_only*]

Create a replication job definition.

evpsn The evpsn option [evpsn=yes|evpsn=no] enables or disables Enterprise Vault partition secure notifications.

metadata_only The metadata_only option [metadata_only=yes|metadata_only=no] enables or disables metadata only replication. This feature is not supported with consistency groups and tunables.

`episodic job show` [*job_name*]

Display the definition of a job.

`episodic job enable` *job_name*

Activate a episodic job and start replicating the data.

`episodic job status` [*job_name*]

Show status of job(s).

`episodic job pause` *job_name*

Pause an already running job.

`episodic episodic job resume` *job_name*

Resume an already paused job.

`episodic episodic job disable` *job_name*

Disable an Enabled/Paused/Failed job.

`episodic job destroy` *job_name* [*option*]

Destroy a job. The administrator can use the force option to clean up local jobs. In case of forceful job removal, configuration of the other clusters, which are part of the job, will not be modified and repunit(s) will be disassociated from the job. Job destroy force should be used in case of unrecoverable configuration or replication direction mismatch.

`episodic job abort` *job_name*

Abort an already running job.

`episodic job sync` *job_name*

Start a sync job. This command is similar to the job enable command but will stop the replication job after one iteration (full or incremental) is complete. This command can also be used if the primary file system is completely destroyed and needs to be recovered from the secondary site after a disaster.

`episodic job failover` *force job_name* [*link_name*]

Failover the job from the source cluster to one of the available target clusters of link *link_name*. Job failover can be executed from the target with empty *link_name* when the source cluster is not available. If the failover command executed from the target (when the source is not available), configuration related to the old source will be deleted. The administrator can use the job failover command from this old source with link_name of the new source to configure left-over source as the target of the current new source.

force The force option [force=yes|force=no] gives the option of interactive failover.

`episodic job failback` *force job_name* [*link_name*]

Failback the job from the source (previous target) cluster to the target (previous source) cluster. link_name is the link to the target cluster.

force The force option [force=yes|force=no] gives the option of interactive failback.

`episodic job exclude` *job_name exclunit_name*

Add an excluding unit to a job.

`episodic job modify` *debug job_name on|off*

Enable or disable debugging for the given job.

`episodic job modify` *tunables job_name netconn rwcount*

Set the tunables (sockets and threads) for the given job.

Display the bandwidth limit value of the replication links.

`continuous enable` *fs_name pool_name link_name*:

Configure the continuous replication between the source and the target cluster. It requires *fs_name*, which should be present on the source cluster, pool_name, which should be present on the target cluster, and link_name, which is created during authentication of the source and the target cluster.

`continuous disable` *fs_name link_name*:

Unconfigure continuous replication from the source and the target cluster. It requires *fs_name* and *link_name* as parameters.

`continuous start` *fs_name*:

Start data replication between the source and the target cluster.

`continuous stop` *fs_name*:

Stop data replication between the source and the target cluster.

`continuous pause` *fs_name*:

Pause data replication between the source and the target cluster.

`continuous resume` *fs_name*:

Resume data replication between the source and the target cluster, which was paused.

`continuous status` *fs_name*:

Display the status of synchronous replication between the source and the target cluster.

`continuous failover` *fs_name*:

Failover the synchronous replication from the source cluster to target cluster. The continuous failover can be executed from the target cluster when the source cluster is not available. If the failover command executed from the target (when the source is not available), it will run the application on the target cluster.

`continuous failback` *fs_name*:

Failback the syncronous replication from the source (previous target) cluster to the target (previous source) cluster.

`continuous show`

Show the list of file system which are configured under synchronous replication.

`continuous service` start

Start the continuous replication service.

`continuous service` stop

Stop the continuous replication service.

`continuous service` status

Check the status of the continuous replication service.

15.1.4 SEE ALSO

episodic(1), continuous(1)

15.2 continuous

15.2.1 SYNOPSIS

`continuous config bind` *ip_addr* [*device*] [*netmask*]

`continuous config show` [*ip* | *remote_clus*]

`continuous config unbind` *ip_addr*

`continuous config export_keys` [*URL*]

`continuous config import_keys` [*URL*]

`continuous config del_keys` *conIP*

`continuous config auth` *conIP link_name*

`continuous config deauth` *link_name*

`continuous service` start [*node_name*]

`continuous service` stop

`continuous service` status

`continuous enable` *fs_name pool_name link_name*

`continuous disable` *fs_name link_name*

`continuous start` *fs_name*

`continuous stop` *fs_name*

`continuous pause` *fs_name*

`continuous resume` *fs_name*

`continuous status` *fs_name*

`continuous failover` *fs_name*

`continuous failback` *fs_name*

`continuous show`

15.2.2 DESCRIPTION

The replication `continuous` command provides synchronous replication between the source and the target cluster, which is based on volume-level replication.

Continuous replication replicates the application writes on the file system at the source location to one remote location across any distance. It provides a consistent copy of application data at the remote location. If a disaster occurs at the source location, you can use the copy of the application data at the remote location and restart the application at the remote location. Continuous replication provides zero RPO.

15.2.3 OPTIONS

`continuous config bind` ***ip_addr*** **[*device*] [*netmask*]** Add a virtual IP address which is used as the continuous replication service virtual IP address and a *device* which is used as a network interface for the virtual IP address. The *ip_addr* is highly-available in the cluster. By default, the replication

virtual IP address is configured on the pubeth0 interface using the pubeth0 interface netmask. You can change the default interface and the netmask using the [*device*] and the [*netmask*] parameters.

continuous config show [*ip* | *remote_clus*] Display the replication virtual IP (using the `ip` option) of the local cluster. The `remote_clus` option of the `continuous config show` command displays information related to the remote cluster replication VIP as well as information related to key import and authentication status.

continuous config unbind *ip_addr* Delete the virtual IP address from the continuous replication service.

continuous config export_keys [*URL*] Copy the continuous replication public key from the local cluster.

continuous config import_keys [*URL*] Copy the continuous replication public key from the remote cluster to the local cluster.

continuous config del_keys *conIP* Delete the continuous replication public key of a remote cluster identified by *conIP* from the local cluster.

continuous config auth *conIP link_name* Authenticate two clusters mutually to participate in the continuous replication service. The *conIP* is the console IP address of the remote cluster, and *link_name* is the association name between the local cluster and the remote cluster.

continuous config deauth *link_name* Un-authenticate two clusters identified by the association *link_name* from participating in the continuous replication service.

continuous service start [node_name] Start the continuous replication service. You can use the optional parameter "node_name" to start the replication service on a specific node of the cluster. Otherwise the service start command automatically figures out the best suitable node in the cluster and starts the replication service there.

continuous service stop Stop the continuous replication service.

continuous service status Check the status of the continuous replication service.

continuous enable *fs_name pool_name link_name* [*delayed*] Configure the continuous replication between the source and the target cluster. It requires fs_name, which should be present on the source cluster, pool_name, which should be present on the target cluster, link_name, which is created during authentication of the source and the target cluster, and [delayed=yes|delayed=no] parameter which specifies if the continuous replication should happen in delayed mode, it is optional for which the default value if not given is 'no'. It requires the pool on the target cluster to have sufficient storage to create a file system and Storage Replicated Log (SRL) volume. The pool name must be same at the source and the target cluster. The SRL volume size is one-fourth of the file system size. If delayed mode is enabled, there can be non-zero RPO. For example, if the file system size on the source cluster is 8GB then at least 3GB storage should be present in the pool at the source cluster to create SRL volume. For the target cluster, 11GB storage should be present in pool to create file system and SRL volume.

continuous disable *fs_name link_name* Unconfigure continuous replication from the source and the target cluster. It requires *fs_name* and *link_name* as parameters.

continuous start *fs_name* Start data replication between the source and the target cluster.

continuous stop *fs_name* Stop data replication between the source and the target cluster.

continuous pause *fs_name* Pause data replication between the source and the target cluster.

continuous resume *fs_name* Resume data replication between the source and the target cluster, which was paused.

continuous status *fs_name* Display the status of continuous replication between the source and the target cluster.

continuous failover *fs_name* Failover the continuous replication from the source cluster to target cluster. There are 2 scenarios in failover case, planned and unplanned failover. In the case of an unplanned failover, this command can be executed from the target cluster when the source cluster is not available. If the failover command executed from the target (when the source is not available), it will run the application on the target cluster and the target cluster will become the new source cluster. In the case of a planned failover, this command can be executed from the source cluster when both the source cluster and the target cluster are reachable. This command switches the roles of the source and the target cluster in continuous replication.

continuous failback *fs_name* Failback the continuous replication. There are 2 scenarios in failback case, planned and unplanned failback. In the case of an unplanned failback, this command can be executed from the original source cluster which was become unavailable due to some reasons. This command will make the original source cluster as new target cluster. In the case of a planned failback, this command can be executed from the target cluster when both the source and target cluster are reachable. It switches the roles of the source and the target cluster in continuous replication.

continuous show Show the list of file system which are configured under continuous replication.

15.2.4 EXAMPLES

Add a VIP address for the replication service.

```
Replication> continuous config bind 192.168.10.56
Please wait...
ACCESS replication SUCCESS V-288-0 192.168.10.56 configured as Virtual IP.
ACCESS replication SUCCESS V-288-0 IP bind completed
```

Configure a VIP address for the replication service on interface pubeth1 and netmask 255.255.255.0.

```
Replication> continuous config bind 192.168.10.56 pubeth1 255.255.255.0
Please wait...
ACCESS replication SUCCESS V-288-0 192.168.10.56 configured as Virtual IP.
ACCESS replication SUCCESS V-288-0 IP bind completed
```

Show the replication VIP address.

```
Replication> continuous config show ip
Local cluster details:
======================
Continuous Replication VIP            : 192.168.10.56
Online On Node                  : isaA_01
```

Delete the replication VIP from the replication service.

```
Replication> continuous config unbind 192.168.10.57
Please wait...
ACCESS replication SUCCESS V-288-0 IP unbind completed.
```

Export the replication key of the local cluster to the remote cluster.

```
Replication> continuous config export_keys

ACCESS replication SUCCESS V-288-1559 Displaying replication key. Please use this key␣
↪with config import_keys command.
```

```
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAtI4Sy9CN7GM0ppfL4NH+bit0/RBosmYadmLelD56li1//
↪CLvYJ28ouC5uuYAx/XnCwCFBBGL/uL3+Sqd5+D8+YaRE0cdJvXJEQ2rwU2Ffm+l7ZX1/
↪uAbhnrRCS8HvuYWm25FgwpYN+6ssDbTDIJKse4M3iRCSKA4bAS9fh/
↪8guCW9yUv1DU1jHIifTATHTEZFk8mesNpq8nOoSxP88vPPAvNde7CBCvWAUERxZQe7CSW5IODT0p6ft0sFR0hBsPs8jqZ61hRze
↪XJyuXx3kqiH7cIYj9x/Rmd2lU9vw== root@isaA_01::isaA_192.168.10.21


ACCESS replication SUCCESS V-288-0 Config export keys command completed successfully.
```

Import the key of a remote cluster.

```
Replication> continuous config import_keys

Enter replication key of remote cluster::ssh-rsa␣
↪AAAAB3NzaC1yc2EAAAABIwAAAQEArkGIdvLGKtxblMVSZHaxxe+vkLtH4tsdQjFrfINY064kQ7lb+MIUhH11Uq/
↪uT4xF1wFXGr2s9yN3pphZkmiABX37VzGm3sXmDllouSHXQ+sqJRiOOBnp6TU6COzv1S8UZxqEWGCYWF1RmpDb2r/
↪6AhL5Mw0DrxfvPRLiCsWdki3iieFyyHaRoXZNnolH5jrn/PZwR9rS1m9QfQaGh3/
↪o8sWE1rphQxLX0D7FyuE0tdfWv4CBpIxFD7gckCx0cSiTG5Ar4ZlpXTSTg81J/
↪6cx46Nwswo8dkknci+YdbhTKjFYdMXw8IRo8aJSwa82B0XmextLbJbqIZ1iVbYpwJDZjw== root@isaB_
↪02::isaB_192.168.10.22

Enter console IP address of remote cluster::192.168.10.22
ACCESS replication SUCCESS V-288-0 Config import keys command completed successfully.
```

Delete the key of a remote cluster.

```
Replication> continuous config del_keys 192.168.10.21
ACCESS replication SUCCESS V-288-0 Config delete keys command completed successfully.
```

Show the list of cluster(s) whose key is imported by the local cluster.

```
Replication> continuous config show

Link name        Remote Console IP   Remote Replication VIP   Time of Key Import      ␣
↪      Time of Authorization
==============   =================   =====================  ␣
↪============================  =====================
--               192.168.10.22       --                       Thu May 25 13:31:55 EDT␣
↪2017
```

Authorize a remote cluster having console IP 192.168.10.22 and link name as rep1.

```
Replication> continuous config auth 192.168.10.22 rep1
Please wait...
ACCESS replication SUCCESS V-288-0 Config auth command completed successfully.
```

Unauthorise a remote cluster having link name as rep1.

```
Replication> continuous config deauth rep1
Please wait...
ACCESS replication SUCCESS V-288-0 Config deauth command completed successfully.
```

Show the list of authorized clusters.

```
Replication> continuous config show
```

```
Link name        Remote Console IP   Remote Replication VIP   Time of Key Import
↪      Time of Authorization
==============   =================   =====================
↪============================  =====================
rep1             192.168.10.22       192.168.10.57            Thu May 25 13:31:55 EDT
↪2017  Thu May 25 13:34:50 EDT 2017
```

Start the replication service.

```
Replication> continuous service start
Starting replication service on isaA_01. Please wait...
ACCESS replication SUCCESS V-288-0 Continuous Replication service started
```

Stop the replication service.

```
Replication> continuous service stop
ACCESS replication SUCCESS V-288-0 Replication service stopped
```

Check the status of the replication service.

```
Replication> continuous service status
Status            : RUNNING
Online On Node    : isaB_01 isaB_02
```

Enable continuous replication.

```
Replication> continuous enable fs1 pool1 rep1
ACCESS Sync_rep SUCCESS V-288-0 Continuous replication configured successfully
```

Disable continuous replication.

```
Replication> continuous disable fs1 rep1
ACCESS Sync_rep SUCCESS V-288-0 Continuous replication unconfigured successfully
```

Start continuous data replication.

```
Replication> continuous start fs1
ACCESS Sync_rep SUCCESS V-288-0 Continuous data replication started successfully
```

Stop continuous data replication.

```
Replication> continuous stop fs1
ACCESS Sync_rep SUCCESS V-288-0 Continuous data replication stopped successfully
```

Pause continuous data replication.

```
Replication> continuous pause fs1
ACCESS Sync_rep SUCCESS V-288-0 Continuous data replication paused successfully
```

Resume continuous data replication.

```
Replication> continuous resume fs1
ACCESS Sync_rep SUCCESS V-288-0 Continuous data replication resumed successfully
```

Display the status of continuous data replication.

```
Replication> continuous status fs1
Name                    value
=====================  =======================
Replicated Data Set     rvg_fs1
Replication Role        Primary
Replication link        rep1

Primary Site Info:

Host name               192.168.10.56
RVG state               enabled for I/O

Secondary Site Info:

Host name               192.168.10.57
Configured mode         synchronous-override
Data status             consistent, up-to-date
Replication status      replicating (connected)
Current mode            synchronous
Logging to              SRL
Timestamp Information   behind by  0h 0m 0s
```

Failover the continuous replication corresponding to the given fs_name.

```
For unplanned failover scenario,

Replication> continuous failover fs1
ACCESS Sync_rep SUCCESS V-288-0 Failover operation executed successfully!

For planned failover scenario,

Replication> continuous failover fs1
ACCESS Sync_rep SUCCESS V-288-0 Primary and secondary roles migrated successfully!
```

Failback the continuous replication corresponding to the given fs_name.

```
For unplanned failback scenario,

Replication> continuous failback fs1
ACCESS Sync_rep SUCCESS V-288-0 Failback operation executed successfully!

For planned failback scenario,

Replication> continuous failback fs1
ACCESS Sync_rep SUCCESS V-288-0 Planned failback executed successfully!
```

Show the list of file system which are configured under continuous replication.

```
Replication> continuous show
Continuous Rep FS Names
=================
fs1
```

15.3 episodic

15.3.1 SYNOPSIS

`episodic config bind` *ip_addr* [*device*] [*netmask*]

`episodic config show` [*ip* | *remote_clus*]

`episodic config unbind` *ip_addr*

`episodic config export_keys` [*URL*]

`episodic config import_keys` [*URL*]

`episodic config del_keys` *conIP*

`episodic config auth` *conIP link_name*

`episodic config deauth` *link_name*

`episodic config check` *link_name*

`episodic config certificate generate`

`episodic config certificate import`

`episodic config certificate delete`

`episodic config certificate status`

`episodic exclunit create` *exclunit_name exclunit_entry* [, *exclunit_entry*,...]

`episodic exclunit add_entry` *exclunit_name exclunit_entry*

`episodic exclunit remove_entry` *exclunit_name exclunit_entry*

`episodic exclunit modify_entry` *exclunit_name old_exclunit_entry new_exclunit_entry*

`episodic exclunit destroy` *exclunit_name*

`episodic exclunit show` [*exclunit_name*]

`episodic job create` *job_name src_repunit tgt_repunit link_name schedule_name* [*evpsn*] [*metadata_only*]

`episodic job show` [*job_name*]

`episodic job enable` *job_name*

`episodic job sync` *job_name*

`episodic job status` [*job_name*]

`episodic job pause` *job_name*

`episodic job resume` *job_name*

`episodic job disable` *job_name*

`episodic job destroy` *job_name* [*option*]

`episodic job abort` *job_name*

`episodic job failover` *force=yes/no job_name* [*link_name*]

`episodic job failback` *force=yes/no job_name link_name*

`episodic job exclude` *job_name exclunit_name*

`episodic job modify` *debug job_name on\off*

`episodic job modify` *tunables job_name netconn rwcount*

`episodic repunit create` *repunit_name repunit_entry*[,*repunit_entry,...]*

`episodic repunit add_entry` *repunit_name repunit_entry*

`episodic repunit remove_entry` *repunit_name repunit_entry*

`episodic repunit modify_entry` *repunit_name old_repunit_entry new_repunit_entry*

`episodic repunit destroy` *repunit_name* [*option*]

`episodic repunit show` [*repunit_name*] | [*verbose*]

`episodic schedule create` *schedule_name minute* [*hour*] [*day_of_the_month*] [*month*] [*day_of_the_week*]

`episodic schedule modify` *schedule_name minute* [*hour*] [*day_of_the_month*] [*month*] [*day_of_the_week*]

`episodic schedule show` [*schedule_name*]

`episodic schedule delete` *schedule_name*

`episodic service` start [*node_name*]

`episodic service` stop

`episodic service` status

15.3.2 DESCRIPTION

config

The replication `episodic config` command provides a set of cluster authentication related operations and methods to add and delete a virtual IP with the replication service.

Veritas Access replication authentication strategy is based on RSA key authentication, and both source and destination clusters have to export their replication public keys. Both the source and destination clusters must then import public keys from each other.

After both the source and destination clusters have successfully imported each other's keys, the `episodic config auth` command must be run on the source cluster to complete the authentication between the two clusters. The `episodic config auth` command will check two-way communication between the source and the destination clusters, and then it will authenticate these two clusters to participate in the replication service.

exclunit

The replication `episodic exclunit` command provides operations to create, destroy, show, and manage excluding unit definitions.

An excluding unit is a collection of files that you do not want to replicate. Hence, all files belonging to an excluding unit are not replicated from the source cluster to the destination cluster.

A single excluding unit can span across multiple directories and is common for all file systems. An excluding unit is defined as an ordered set of entries, where each entry is one of the following:

- Directory
- Single file

The entry is identified by the path of the directory, or the file. The entry should be the substring following the file system root. For example, /vx/fs1/dir1/file specifies dir1/file as the exclunit entry. Member entries are ordered inside an excluding unit.

The exclunit entry has higher priority over the repunit entry. If any existing file name matches the exclunit entry, the file/directory is not replicated to the target.

job

The replication `episodic job` command provides operations to create, destroy, display, and manage job definitions and job states.

A job binds together source and target replication units, the source and the target clusters, and a replication schedule. A configured job defines who participates in replication, what gets replicated, and when replication occurs between the source and the target clusters.

repunit

The replication `episodic repunit` command provides operations to create, destroy, show, and manage replication unit definitions. A replication unit is a collection of files that you want to include in a replication job. All files belonging to a replication unit are replicated together from the source cluster to the target cluster.

A single replication unit can span across multiple directories and even multiple file systems. A replication unit is defined as an ordered set of entries, where each entry is one of the following:

- File system
- Subdirectory
- Single file

The entry is identified by the file system name optionally followed by a slash ('/'), followed by the path of the directory, or the file inside the file system. Member entries are ordered inside a replication unit and such ordering information is used for the mapping between the source and the target replication units.

The user can specify a file/directory name to replicate files based on file name, which will select files dynamically. The user does not need to know the exact file name in advance.

The exclunit entry have higher priority over the repunit entry. If any path name matches the exclunit entry or the exclunit the file/directoryis not replicated to the target. Also, if any path matches both the repunit and the exclunit, the path is not replicated to the target. Based on dataset known, the user can specify either the repunit or the exclunit .

schedule

The replication `schedule` commands create, destroy, display, and manage the replication schedules.

Veritas Access supports periodic replication, where the data gets replicated from the source to the destination at regular intervals as defined by the schedule. The same schedule definition can be used for multiple replication jobs. The input to create a schedule is in the same format as that required by the standard UNIX cron daemon, where the values for minute, hour, day-of-the-month, month, and day-of-the-week, are almost the same as the crontab format. The parameters and values are defined below.

1. `Schedule name`: Specifies the name of the schedule.

2. `Minute`: This parameter contains either an asterisk '*', which implies run every minute, or a numeric value between the range of 0-59.

3. `Hour`: This parameter contains either an asterisk '*', which implies run every hour, or a numeric value between the range of 0-23.

4. `Day of month`: This parameter contains either an asterisk '*', which implies run every day of the month, or a numeric value between the range of 1-31.

5. `Month:` This parameter contains either an asterisk '*', which implies run every month, or a numeric value between the range of 1-12. In addition, this field accepts names of the month. Enter the first three letters of the month (not case-sensitive) as the input value.

6. `Day of Week:` This parameter contains either an asterisk '*', which implies run every day of the week, or a numeric value between the range of 0-6, with 0 interpreted as Sunday. In addition, this field accepts names of the week. Enter the first three letters of the month (only use lowercase letters) as the input value.

service

The replication `episodic service` commands start, stop, and display the status of the replication service. Before starting the Veritas Access Replication service, make sure that:

1. A file system is present on the cluster.
2. A virtual IP address is added to the replication service. Refer to the config part of this man page.

15.3.3 OPTIONS

`episodic config bind` ***ip_addr*** **[*device*] [*netmask*]** Add a virtual IP address to be used as the replication service virtual IP address and *device* as a network interface for the virtual IP address. The *ip_addr* is highly-available in the cluster. By default, the replication virtual IP address is configured on the pubeth0 interface using the pubeth0 interface netmask. You can change the default interface and the netmask using [*device*] and [*netmask*] parameters.

`episodic config show` **[*ip* | *remote_clus*]** Display the replication virtual IP (using the `ip` option) of the local cluster `.remote_clus` option of the `episodic config show` command displays information related to the remote cluster replication VIP as well as information related to key import and authentication status.

`episodic config unbind` ***ip_addr*** Delete the virtual IP address from the replication service.

`episodic config export_keys` **[*URL*]** Copy the replication public key from the local cluster to a specified URL.

`episodic config import_keys` **[*URL*]** Copy the replication public key of the remote cluster from a specified URL to the local cluster.

`episodic config del_keys` ***conIP*** Delete the replication public key of a remote cluster identified by *conIP* from the local cluster.

`episodic config auth` ***conIP link_name*** Authenticate two clusters mutually to participate in the replication service. The *conIP* is the console IP address of the remote cluster, and *link_name* is the association name between the local cluster and the remote cluster.

`episodic config deauth` ***link_name*** Un-authenticate two clusters identified by the association *link_name* from participating in the replication service.

`episodic config check` ***link_name*** Check the status of the association identified by the name *link_name* between the source and the destination clusters.

`episodic config certificate generate` Generate a self-signed certificate.

`episodic config certificate import` Import trusted certificates from an external entity.

`episodic config certificate delete` Delete configured certificates.

`episodic config certificate status` Display certificate information.

episodic exclunit create ***exclunit_name exclunit_entry*** **[,** ***exclunit_entry*****,...]** Create an excluding unit with a given set of entries.

episodic exclunit add_entry ***exclunit_name exclunit_entry*** Add a new entry to an existing excluding unit.

episodic exclunit remove_entry ***exclunit_name exclunit_entry*** Remove an entry from an existing excluding unit.

episodic exclunit modify_entry ***exclunit_name old_exclunit_entry new_exclunit_entry*** Modify an existing entry in a excluding unit.

episodic exclunit destroy ***exclunit_name*** Destroy an excluding unit.

exclunit show [***exclunit_name*****]** Display excluding unit details.

`episodic job create` *job_name src_repunit tgt_repunit link_name schedule_name* [*evpsn*] [*metadata_only*]

Create a replication job with source replication unit, target replication unit and the link name. The evpsn option [evpsn=yes|evpsn=no] enables or disables Enterprise Vault partition secure notifications. The metadata_only option [metadata_only=yes|metadata_only=no] enables or disables metadata only replication. This feature is not supported with consistency groups and tunables.

`episodic job show` [*job_name*]

Display the definition of a job.

`episodic job enable` *job_name*

Start a job and add it to the replication scheduler queue. The actual replication of data starts on the next schedule of the job.

`episodic job status` [*job_name*]

Display status of job(s).

`episodic episodic job pause` *job_name*

Pause a running job.

`episodic job resume` *job_name*

Resume a paused job.

`episodic job disable` *job_name*

Disable an enabled job.

`episodic job destroy` *job_name* [*option*]

Destroy a job. The administrator can use the force option to clean up the local job. In case of forceful job removal, configuration of the other clusters, which are part of the job, are not modified and repunit(s) are disassociated from the job. Job destroy force should be used in case of unrecoverable configuration or replication direction mismatch.

`episodic job abort` *job_name*

Abort a running job.

`episodic job sync` *job_name*

This command is similar to the job enable command but will stop the replication job after one iteration (full or incremental) is complete. This command can also be used if the primary file system is completely destroyed and needs to be recovered from the secondary site after a disaster.

`episodic job failover` *force job_name* [*link_name*]

Failover the job from the source cluster to one of the available target clusters of link *link_name*. Job failover can be executed from the target with empty *link_name* when the source cluster is not available. If the failover command executed from the target (when the source is not available), configuration related to the old source is deleted. The administrator can use the job failover command from this old source with the link_name of the new source to configure the leftover source as the target of the current new source.

`episodic job failback` *force job_name link_name*

Failback the job from the source (previous target) cluster to the target (previous source) cluster. link_name is the link to the target cluster.

`episodic job exclude` *job_name exclunit_name*

Add excluding unit to a specified job. exclunit_name is the excluding unit name to add for job job_name.

`episodic job modify` *debug job_name on\off*

Enable or disable debugging for the given job.

`episodic job modify` *tunables job_name netconn rwcount*

Set the tunables (sockets and threads) for the given job.

`episodic repunit create` ***repunit_name repunit_entry*** **[,** ***repunit_entry*****,...]** Create a replication unit with a given set of entries.

`episodic repunit add_entry` ***repunit_name repunit_entry*** Add a new entry to an existing replication unit.

`episodic repunit remove_entry` ***repunit_name repunit_entry*** Remove an entry from an existing replication unit.

`episodic repunit modify_entry` ***repunit_name old_repunit_entry new_repunit_entry*** Modify an existing entry in a replication unit.

`episodic repunit destroy` ***repunit_name*** **[*****option*****]** Destroy a replication unit. With force option, repunit is removed even if the associated job exists.

`episodic repunit show` **[*****repunit_name*****] | [*****verbose*****]** Display replication unit details.

Display compact list if no option is provided.

Use 'verbose' keyword to display all repunit details including paths.

`episodic schedule create` ***schedule_name minute*** **[*****hour*****] [*****day_of_the_month*****] [*****month*****] [*****day_of_the_week*****]** Create a schedule.

`episodic schedule modify` ***schedule_name minute*** **[*****hour*****] [*****day_of_the_month*****] [*****month*****] [*****day_of_the_week*****]** Modify an existing schedule.

`episodic schedule show` **[*****schedule_name*****]** Display a schedule.

`episodic schedule delete` ***schedule_name*** Delete a schedule.

`episodic service` **start [`node_name`]** Start the replication service. You can use the optional parameter "node_name" to start the replication service on a specific node of the cluster. Otherwise the service start command automatically figures out the best suitable node in the cluster and starts the replication service there.

`episodic service` **stop** Stop the replication service.

`episodic service` **status** Check the status of the replication service.

15.3.4 EXAMPLES

config

Add a VIP address for the replication service.

```
    Replication> episodic config bind 10.209.107.89
Please wait...
Completed
```

Configure VIP address for the replication service on interface pubeth1 and netmask 255.255.248.0.

```
Replication> episodic config bind 10.209.107.89 pubeth1 255.255.248.0
Please wait...
Completed
```

Show the replication VIP address.

```
Replication> episodic config show ip
Replication VIP :: 10.209.107.89
```

Delete the replication VIP from the replication service.

```
Replication> episodic config unbind 10.209.107.89
Please wait...
Completed
```

Export replication key of the local cluster to the home directory of fBssnasfp user at URL `ssnasweb.vxindia.veritas.com`.

```
Replication>  episodic config export_keys scp://ssnas@ssnasweb.vxindia.veritas.com:~/
Password: ******
ACCESS Replication SUCCESS V-288-0 Key file ISAKEY_source_10.209.105.128_2009-05-29
copied successfully at location ssnasweb.vxindia.veritas.com:~/
```

Import key of a remote cluster from the home directory of `ssnas` user at URL `ssnasweb.vxindia.veritas.com`.

```
Replication>  episodic config import_keys scp://ssnas@ssnasweb.vxindia.veritas.com:~/
↪ISAKEY_source_10.209.105.236_2009-05-29
Password: ******
```

Show list of cluster(s) whose key is imported by the local cluster.

```
Replication>  episodic config show remote_clus
Link name   Remote cluster ConsoleIP   Remote cluster Replication IP   Time of Key␣
↪Import             Time of Authorization
=========   ========================   ============================  ␣
↪============================   =====================
--          10.209.105.236             --                              Wed November 7␣
↪19:22:47 UTC 2014   --
```

Authorize a remote cluster having console IP 10.209.105.236 and link name as Pune_Shanghai.

```
Replication>  episodic config auth 10.209.105.236 Pune_Shanghai
```

Show list of authorized clusters.

```
Replication>  config show remote_clus
Link name     Remote cluster ConsoleIP   Remote cluster Replication IP   Time of Key␣
↪Import            Time of Authorization
=========     ========================   ==============================  ␣
↪============================  ============================
Pune_Shanghai 10.209.105.236             10.209.107.90                   Wed November␣
↪7 19:22:47 UTC 2014   Wed November  7 22:09:46 UTC 2014
```

exclunit

Create an excluding unit `exclu1` with entries `DB_CONFIG`, `SQL_CONFIG`, and `DB_DIR/conf_file`.

```
Replication> exclunit create exclu1 DB_CONFIG,SQL_CONFIG,DB_DIR/conf_file
ACCESS Replication SUCCESS V-288-0 Excluding unit exclu1 created successfully.
```

Display the definition of the exclunit `exclu1`.

```
Replication>episodic exclunit show exclu1

exclunit name: exclu1
======================

fs name:
--------

paths:
------
DB_CONFIG
SQL_CONFIG
DB_DIR/conf_file

patterns:
---------

jobs:
-----
new_job1
```

Add an entry `DB_DIR/new_conf_file` in the excluding unit definition **exclu1**.

```
Replication> episodic exclunit add_entry exclu1 DB_DIR/new_conf_file
```

Delete an entry `DB_DIR/new_conf_file` from the excluding unit definition **exclu1**.

```
Replication> episodic exclunit remove_entry exclu1 DB_DIR/new_conf_file
```

Modify an entry `DB_DIR/new_conf_file` to `DB_DIR/new_conf_file_1` in the excluding unit definition **exclu1**.

```
Replication> episodic exclunit modify_entry exclu1 DB_DIR/new_conf_file DB_DIR/new_
↪conf_file_1
```

job

Create a job with name `new_job1`, source replication unit `srp1`, target replication unit `trp1`, link name as `Pune_Shanghai`, and schedule as `s1`.

```
Replication> episodic job create new_job1 srp1 trp1 Pune_Shanghai s1
Completed successfully.
```

Display definition of job `new_job1`:

```
Replication> episodic job show new_job1

Job Name              Role      Debug   Schedule   State       Exclunit   Source␣
↪repunit   Target repunit   Link name
========              ====      =====   ========   ========    ========  ␣
↪==============   ==============   ============
new_job1              SOURCE    ON      sch1       ENABLED     --         srp1       ␣
↪    trp1             Pune_Shanghai
```

Check status of a job `new_job1`:

```
Replication>  episodic job status new_job1
Job 'new_job1' configured but not running.
```

Enable a job `new_job1`:

```
Replication> episodic job enable new_job1
```

Pause a running job `new_job1`:

```
Replication>  episodic job pause new_job1
```

Resume a paused job `new_job1`:

```
Replication>  episodic job resume new_job1
```

Disable an enabled job `new_job1`:

```
Replication> episodic job disable new_job1
```

Destroy a disabled job `new_job1`:

```
Replication> episodic job destroy new_job1
```

Abort a running job `new_job1`.

```
Replication> job abort new_job1
```

Start sync on a disabled job `new_job1`.

```
Replication> episodic job sync new_job1
```

Failover the job `new_job1` corresponding to the link `Pune_Shanghai` to target cluster. In planned failover scenarios, the administrator can use this command.

```
Replication> episodic job failover force=no new_job1 Pune_Shanghai
```

Failback the job `new_job1` to the source cluster from the target cluster.

```
Replication> episodic job failback force=no new_job1 Pune_Shanghai
```

Add excluding unit `exclu1` to job `new_job1`:

```
Replication> episodic job exclude new_job1 exclu1
```

Modify the job with tunables.

```
Replication> episodic job modify tunables new_job1 4 1
```

repunit

Create a replication unit `ru1` with entries `fs1/DB_CONFIG`, `fs1/SQL_CONFIG` and `fs1/conf_file`.

```
Replication> episodic repunit create ru1 fs1/DB_CONFIG,fs1/SQL_CONFIG,fs1/conf_file
```

Display the definition of a repunit `ru1`.

```
Replication> episodic repunit show ru1

repunit name: ru1
===================

fs name:
--------
fs1,fs2

paths:
------
fs1/DB_CONFIG
fs1/SQL_CONFIG
fs1/conf_file
fs2/conf_file

patterns:
---------

jobs:
-----
new_job1
```

Add an entry `fs1/new_conf_file` in the replication unit definition `ru1`.

```
Replication> repunit add_entry ru1 fs1/new_conf_file,fs2/new_config_dir1
```

Delete an entry `fs1/new_conf_file` from the replication unit definition `ru1`.

```
Replication> episodic repunit remove_entry ru1 fs1/new_conf_file
```

Modify an entry `fs1/new_conf_file` to `fs1/new_conf_file_1` in the replication unit definition `ru1`.

```
Replication> episodic repunit modify_entry ru1 fs1/new_conf_file fs1/new_conf_file_1
```

schedule

Create a schedule `s1` to run a job every 30 minutes.

```
Replication> episodic schedule create s1 */30
```

Display schedule `s1`.

```
Replication> schedule show s1
Schedule Name                Minute     Hour       Day        Month      WeekDay
===============              ======     =====      ======     ======     ========
s1                           */30      *          *          *          *
```

Modify a schedule `s1` to run a job every 40 minutes.

```
Replication> episodic schedule modify s1 40
```

Delete a schedule `s1`.

```
Replication> episodic schedule delete s1
```

service

Start the replication service.

```
Replication> episodic service start
```

Check the status of the replication service.

```
Replication> episodic service status
Replication service RUNNING on node nasg01_01.
```

Stop the replication service.

```
Replication> episodic service stop
```

16 SmartIO Commands

16.1 SmartIO

16.1.1 SYNOPSIS

`cache create` *node_name disk size fstype cachename*

`cache delete` *node_name cache_name*

`cache grow` *cache_size node_name cache_name*

`cache list`

`cache offline` *node_name cache_name*

`cache online` *node_name cache_name*

`cache shrink` *cache_size node_name cache_name*

`cache stat` *node_name*

`device list` *node_name*

`fs cachemode` {`nocache`|`read`|`writeback`} *fs_name*

`fs disable` *fs_name*

`fs enable` *fs_name*

`fs flush` *fs_name*

`fs list` *fs_name*

`fs purge` *fs_name*

`fs stat` *fs_name*

`file cachemode` {`nocache`|`read`|`writeback`} *filename*

`file list` *filename*

`file load` *filename*

`file pin` *filename*

`file restore_access` *filename*

`file unpin` *filename*

16.1.2 DESCRIPTION

These `SmartIO` commands are used to create and manage SmartIO Cache. Veritas Access supports read caching on solid-state drives (SSDs) for applications running on Veritas Access file systems. In this scenario, application reads are satisfied from the cache whenever possible. As the application accesses the file system, the file system loads data from the disk into the cache. Application writes go to the disk in the usual way. With each write, the file system synchronizes the cache to ensure that applications never see stale data. If a cache device fails, a file that is cached in read mode is completely present on the disk. Therefore, the cache failure does not affect the application I/Os for the file and the application I/Os continue without interruption. By default, the cache area is enabled for caching. All file systems on the system are cached unless you explicitly disable caching for that file system. You do not need to explicitly enable caching on a file system.

These commands allow creating SmartIO cache, setting cache policies for the file systems files and view the SmartIO cache statistics. The supported SmartIO cache policies are read, writeback, and nocache. These commands also allow you to pin and unpin files and directories to the SmartIO cache. When a policy is set for a directory, all the files under the directory inherit the cache policy. The SmartIO cache is local to the node where it is created. Make sure that you create the SmartIO cache on each node of the Veritas Access cluster.

SmartIO provides write caching in the writeback mode also. In writeback mode, an application write returns success after the data is written to the SmartIO cache, which is usually on an SSD. At a later time, SmartIO flushes the cache, which writes the dirty data to the disk. Writeback caching expects to improve the latencies of synchronous user data writes. Writeback caching is superset of read caching. When writeback caching is enabled, read caching is implicitly enabled. Reads are satisfied from the cache if possible, and the file system transparently loads file data into the cache. Both read and writeback caching may be enabled for the same file at the same time. The writeback caching mode gives good performance for writes, but also means that the disk copy may not always be up to date. If a cache device fails, a file that is cached in writeback mode may not be completely present on the disk. SmartIO has a mechanism to flush the data from the cache device when the device comes back online. The writeback cache policy is supported on a two-node Veritas Access clusters only. Veritas Access supports multiple cache creations.

16.1.3 OPTIONS

`cache create` ***node_name disk size fstype cachename*** Create the SmartIO cache on the *cache_name* using the specified SSD disk.

`cache delete` ***node_name cache_name*** Delete the specified SmartIO cache on the the specified node.

`cache grow` ***cache_size node_name cache_name*** Grow the specified SmartIO cache on the specified node. The `cache_size` is specified in GB or TB.

`cache list` List the SmartIO cache properties on the specified node.

`cache offline` ***node_name cache_name*** Bring the SmartIO cache offline on the specified node.

`cache online` ***node_name cache_name*** Bring the SmartIO cache online on the specified node.

`cache shrink` ***cache_size node_name cache_name*** Shrink the SmartIO cache on the specified node. The `cache_size` is specified in GB or TB.

`cache stat` ***node_name*** Display the SmartIO cache statistics on the specified node

`device list` ***node_name*** List the SSD devices available on *node_name*.

`fs cachemode` ***{nocache | read | writeback} fs_name*** Set `read`, `nocache` or `writeback` SmartIO cache policy for `fs_name` file system.

fs disable *fs_name* Disable SmartIO cache for the `fs_name` file system. This action stops SmartIO cache for the file system specified.

fs enable *fs_name* Enable SmartIO cache for the `fs_name` file system. This action starts caching for the specified file system.

fs flush *fs_name* Flush the writeback data for the `fs_name` file system. For this command to work, the file system must be configured with `writeback` cache.

fs list *fs_name* Display the SmartIO cache properties and usage for the `fs_name` file system.

fs purge *fs_name* Purge or remove the cached contents for the `fs_name` file system in SmartIO cache.

fs stat *fs_name* Display detailed SmartIO cache statistics for the `fs_name` file system specified.

file cachemode *{nocache|read|writeback} file_name* Set caching mode for the named file or directory. The supported caching modes are read, nocache or writeback. For directories, caching mode is set recursively.

file load *file_name* Load the named file or directory into the smartIO cache. The file is loaded asynchronously in the background.

file pin *file_name* Pin the named file or directory into SmartIO cache. The pinned file blocks in the cache generally will not be aged out of the cache.

file list *file_name* List the SmartIO cache properties for the specified `file_name`.

file restore_access *file_name* Enables access to files that are missing writeback data. If the writeback cache data could not be flushed to the disk for some files due to an error, SmartIO prevents access to the files. This command only restores access to the files, but it does not restore the missing data.

file unpin *file_name* Unpin the named file or directory into smartIO cache.

16.1.4 EXAMPLES

Create a SmartIO cache.

```
test1.SmartIO>cache create sdb,sdc 2G reserve cache6
SmartIO cache area created
```

List SmartIO cache properties.

```
test1.SmartIO> cache list

    NAME            TYPE   SIZE       ASSOC-TYPE  STATE      FSTYPE     DEVICE
====            ====   ====       ==========  =====      ======             ======
cache4          VxFS   2.00g      AUTO        OFFLINE    default    sdb
    cache6          VxFS   2.00g      AUTO        ONLINE     reserve    sdb,sdc
```

Bring the SmartIO cache online.

```
test1.SmartIO> cache online test1_01 cache2
SmartIO cache is online
```

Grow the SmartIO cache.

```
test1.SmartIO> cache grow 16g test1_01 sdd cache2
Smartio cache grow completed
```

Change the fstype of SmartIO cache.

```
test1.SmartIO> cache settype clusA_01 cache5 reserve
ACCESS smartio SUCCESS V-288-999 SmartIO cache fstype change
```

Display SmartIO cache statistics.

```
test1.SmartIO> cache stat

TYPE: VxFS
NAME: sfcachearea_1
               Cache Size:   15.97 GB
        Cache Utilization:   118.0 MB ( 0.72 %)
File Systems Using Cache:       7
Writeback Cache Use Limit:  Unlimited
  Writeback Flush Timelag:      10 s

Read Cache                                  Writeback
Hit Ratio    Data Read    Data Written    Hit Ratio    Data Written

Total:
   8.97 %        52 KB            1 MB       0.00 %            0 KB

/shared:
   8.97 %        52 KB            1 MB       0.00 %            0 KB
```

Set nocache SmartIO policy for file system where TPCC database archivelog files are stored.

```
SmartIO> fs cachemode nocache tpcc_arch1

Cluster-configuration updated with changed Mount Options for
node test_01
Mount Point /vx/tpcc_arch1 remounted successfully on test_01
Cluster-configuration updated with changed Mount Options for
node test_02
Mount Point /vx/tpcc_arch1 remounted successfully on test_02
```

Enable SmartIO cache for file system where TPCC datafiles are stored.

```
SmartIO> fs enable tpcc_data1
```

List SmartIO cache for file system tpcc_data1

```
SmartIO> fs list tpcc_data1
/vx/tpcc_data1:
READ CACHE    WRITEBACK    MODE       PINNED  NAME
    648 KB         0 KB    read       no      /vx/tpcc_data1/cntrlSLOB.ctl
      0 KB         0 KB    read       no      /vx/tpcc_data1/SLOB
    648 KB         0 KB    read       no      /vx/tpcc_data1
```

List SmartIO cache for file system tpcc_data1

```
SmartIO> fs stat tpcc_data1

       Cache Size:   15.97 GB
Cache Utilization:     652 KB ( 0.00 %)

Read Cache
```

```
Hit Ratio    Data Read    Data Written

/vx/tpcc_data1:
  96.40 %     12.55 MB        1.656 MB

/vx/tpcc_data1:
READ CACHE    WRITEBACK    MODE       PINNED  NAME
```

Load system.dbf file into SmartIO cache

```
SmartIO> file load /vx/tpcc_data/system.dbf
SUCCESS: SmartIO cache load for  /vx/tpcc_data1/system.dbf on host ISAGA_01
```

Pin system.dbf file into SmartIO cache

```
SmartIO> file pin /vx/tpcc_data/system.dbf
SUCCESS : SmartIO cache  pin for /vx/tpcc_data1/system.dbf
```

Display SmartIO information for file system.dbf

```
SmartIO> file list /vx/tpcc_data/system.dbf
/vx/tpcc_data1/system.dbf:
READ CACHE    WRITEBACK    MODE       PINNED
 16.68 MB         0 KB    read       yes
```

Display SmartIO cache information for directory /vx/tpcc_data1

```
SmartIO> file list /vx/tpcc_data1
/vx/tpcc_data1:
READ CACHE    WRITEBACK    MODE       PINNED  NAME
  16.68 MB         0 KB    read       no      /vx/tpcc_data1/cntrlSLOB.ctl
  18.26 MB         0 KB    read       no      /vx/tpcc_data1//o1_system.dbf
   14.8 MB         0 KB    read       no      /vx/tpcc_data1//o1_sysaux.dbf
  2.009 GB         0 KB    read       no      /vx/tpcc_data1//o1_mf_iops.dbf
   2.04 GB         0 KB    read       no      /vx/tpcc_data1/SLOB/datafile
      0 KB         0 KB    read       yes     /vx/tpcc_data1/control1.ctl
  16.68 MB         0 KB    read       yes     /vx/tpcc_data1/system.dbf
   2.07 GB         0 KB    read       no      /vx/tpcc_data1
```

16.1.5 SEE ALSO

cache(1), fs(1), file(1)

16.2 cache

16.2.1 SYNOPSIS

`cache create` *node_name disk size fstype cachename*

`cache delete` *node_name cache_name*

`cache grow` *cache_size node_name cache_name*

`cache list`

`cache offline` *node_name cache_name*

`cache online` *node_name cache_name*

`cache shrink` *cache_size node_name cache_name*

`cache stat` *node_name*

`cache wb_size` *node_name cache_name cache_size*

16.2.2 DESCRIPTION

The `SmartIO` cache commands are used to create and manage the SmartIO cache.

The cache commands let you create, delete, or resize the SmartIO cache. You can bring the SmartIO cache online or offline. You can view the statistics of the SmartIO cache usage and list the cache properties. You must specify the `node_name` as one of the arguments, because the SmartIO cache is local to a node. Veritas Access supports multiple cache creations.

16.2.3 OPTIONS

`cache create` ***node_name disk size fstype cachename*** Create the SmartIO cache on the *cache_name* using the specified SSD disk.

`cache delete` ***node_name cache_name*** Delete the specified SmartIO cache on the the specified node.

`cache grow` ***cache_size node_name cache_name*** Grow the specified SmartIO cache on the specified node. The `cache_size` is specified in GB or TB.

`cache list` List the SmartIO cache properties on the specified node.

`cache offline` ***node_name cache_name*** Bring the SmartIO cache offline on the specified node.

`cache online` ***node_name cache_name*** Bring the SmartIO cache online on the specified node.

`cache shrink` ***cache_size node_name cache_name*** Shrink the SmartIO cache on the specified node. The `cache_size` is specified in GB or TB.

`cache stat` ***node_name*** Display the SmartIO cache statistics on the specified node.

`cache wb_size` ***node_name cache_name cache_size*** Set the SmartIO cache writeback size.

16.2.4 EXAMPLES

Create a SmartIO cache.

```
test1.SmartIO>cache create sdb,sdc 2G reserve cache6
SmartIO cache area created
```

List the SmartIO cache properties.

```
test1.SmartIO> cache list

    NAME            TYPE   SIZE       ASSOC-TYPE  STATE      FSTYPE     DEVICE
====            ====   ====       ==========  =====      ======             ======
cache4          VxFS   2.00g      AUTO        OFFLINE    default    sdb
    cache6          VxFS   2.00g      AUTO        ONLINE     reserve    sdb,sdc
```

Bring the SmartIO cache online.

```
test1.SmartIO> cache online test1_01 cache6
SmartIO cache is online
```

Grow the SmartIO cache.

```
test1.SmartIO> cache grow 16g test1_01 cache2
Smartio cache grow completed
```

Change the fstype of SmartIO cache.

```
test1.SmartIO> cache settype clusA_01 cache5 reserve
ACCESS smartio SUCCESS V-288-999 SmartIO cache fstype change
```

Display SmartIO cache statistics.

```
test1.SmartIO> cache stat

TYPE: VxFS
NAME: cache1
               Cache Size:   15.97 GB
        Cache Utilization:   118.0 MB ( 0.72 %)
File Systems Using Cache:       7
Writeback Cache Use Limit:  Unlimited
  Writeback Flush Timelag:      10 s

Read Cache                                Writeback
Hit Ratio    Data Read    Data Written    Hit Ratio    Data Written

Total:
   8.97 %        52 KB            1 MB       0.00 %            0 KB

/shared:
   8.97 %        52 KB            1 MB       0.00 %            0 KB
```

Set the SmartIO cache writeback size.

```
test1.SmartIO> cache wb_size sun_01 cache6 8g
ACCESS smartio SUCCESS V-288-999 SmartIO WB cache set to 8g
```

16.2.5 SEE ALSO

cache(1), fs(1), file(1)

16.3 device

16.3.1 SYNOPSIS

`device list` *node_name*

16.3.2 DESCRIPTION

The `SmartIO` device command is used to list all available SSD devices on *node_name*.

The device command lets you list all available SSD devices on *node_name*. You must specify the `node_name` as one of the arguments, because the SmartIO device is local to a node.

16.3.3 OPTIONS

`device list` *node_name* List the SSD devices available on *node_name*.

16.3.4 EXAMPLES

List SSD Devices:

```
test1.SmartIO> device list test1_01
NODE-NAME =  test1_01
=================


DEVICE          STATUS
======          ======
intel_ssd0_0    online
intel_ssd1_0    online
intel_ssd2_0    online
```

16.3.5 SEE ALSO

cache(1), fs(1), file(1)

16.4 file

16.4.1 SYNOPSIS

`file cachemode {nocache|read|writeback}` *file_name*

`file list` *file_name*

`file load` *file_name*

`file pin` *file_name*

`file restore_access` *file_name*

`file unpin` *file_name*

16.4.2 DESCRIPTION

The `SmartIO` file commands are used to set cache policies, list current cache policies at file or directory. The argument *file_name* refers to a single file or directory name. You can load, pin and unpin specified files into the SmartIO cache. The writeback cache policy is supported on two node Veritas Access clusters only.

16.4.3 OPTIONS

`file cachemode` ***{nocache|read|writeback} file_name*** Set caching mode for the named file or directory. The supported caching mode are read, nocache or writeback. For directories, caching mode is set recursively.

`file load` ***file_name*** Load the named file or directory into the SmartIO cache. The file is loaded asynchronously in background.

`file pin` ***file_name*** Pin the named file or directory into SmartIO cache. The pinned file blocks in the cache generally will not be aged out of the cache.

`file list` ***file_name*** List the SmartIO cache properties for the specified `file_name`.

`file restore_access` ***file_name*** Enables access to files that are missing writeback data. If the writeback cache data could not be flushed to the disk for some files due to an error, SmartIO prevents access to the files. This command only restores access to the files, but it does not restore the missing data.

`file unpin` ***file_name*** Unpin the named file or directory into smartIO cache.

16.4.4 EXAMPLES

Load the **system.dbf** file into the SmartIO cache.

```
SmartIO> file load /vx/tpcc_data/system.dbf
SUCCESS : SmartIO cache  load for /vx/tpcc_data1/system.dbf on host ISAGA_01
```

Pin the **system.dbf** file into SmartIO cache.

```
SmartIO> file pin /vx/tpcc_data/system.dbf
SUCCESS : SmartIO cache  pin for /vx/tpcc_data1/system.dbf
```

Display SmartIO information for file system.dbf.

```
SmartIO> file list /vx/tpcc_data/system.dbf
/vx/tpcc_data1/system.dbf:
READ CACHE    WRITEBACK    MODE       PINNED
  16.68 MB         0 KB    read       yes
```

Display SmartIO cache information for directory **/vx/tpcc_data1**.

```
SmartIO> file list /vx/tpcc_data1
/vx/tpcc_data1:
READ CACHE    WRITEBACK    MODE       PINNED  NAME
  16.68 MB         0 KB    read       no      /vx/tpcc_data1/cntrlSLOB.ctl
  18.26 MB         0 KB    read       no      /vx/tpcc_data1//o1_system.dbf
   14.8 MB         0 KB    read       no      /vx/tpcc_data1//o1_sysaux.dbf
  2.009 GB         0 KB    read       no      /vx/tpcc_data1//o1_mf_iops.dbf
   2.04 GB         0 KB    read       no      /vx/tpcc_data1/SLOB/datafile
      0 KB         0 KB    read       yes     /vx/tpcc_data1/control1.ctl
  16.68 MB         0 KB    read       yes     /vx/tpcc_data1/system.dbf
   2.07 GB         0 KB    read       no      /vx/tpcc_data1
```

16.4.5 SEE ALSO

cache(1), fs(1), file(1)

16.5 fs

16.5.1 SYNOPSIS

`fs cachemode {nocache|read|writeback}` *fs_name*

`fs disable` *fs_name*

`fs enable` *fs_name*

`fs flush` *fs_name*

`fs list` *fs_name*

`fs purge` *fs_name*

`fs stat` *fs_name*

16.5.2 DESCRIPTION

The `SmartIO` file system level commands are used to enable, disable, flush, or purge the SmartIO cache for the file system specified by *fs_name*. The cachemode command lets you set the caching policy for the entire file system. The possible cachemodes are read, nocache, and writeback.

16.5.3 OPTIONS

`fs cachemode` *{nocache | read | writeback} fs_name* Set `read`, `nocache` or `writeback` SmartIO cache policy for `fs_name` file system.

`fs disable` *fs_name* Disable SmartIO cache for the `fs_name` file system. This action stops SmartIO cache for the file system specified.

`fs enable` *fs_name* Enable SmartIO cache for the `fs_name` file system. This action starts caching for the specified file system.

`fs flush` *fs_name* Flush the writeback data for the `fs_name` file system. For this command to work, the file system must be configured with `writeback` cache.

`fs list` *fs_name* Display the SmartIO cache properties and usage for the `fs_name` file system.

`fs purge` *fs_name* Purge or remove the cached contents for the `fs_name` file system in SmartIO cache.

`fs stat` *fs_name* Display detailed SmartIO cache statistics for the `fs_name` file system specified.

16.5.4 EXAMPLES

Set the nocache SmartIO policy for file system where TPCC database archivelog files are stored.

```
SmartIO> fs cachemode nocache tpcc_arch1
Cluster-configuration updated with changed Mount Options for
node test_01
Mount Point /vx/tpcc_arch1 remounted successfully on test_01
Cluster-configuration updated with changed Mount Options for
node test_02
Mount Point /vx/tpcc_arch1 remounted successfully on test_02
```

Enable the SmartIO cache for file system where TPCC data files are stored.

```
SmartIO> fs enable tpcc_data1
```

List the SmartIO cache for file system `tpcc_data1`.

```
SmartIO> fs list tpcc_data1
/vx/tpcc_data1:
READ CACHE    WRITEBACK    MODE       PINNED  NAME
    648 KB         0 KB    read       no      /vx/tpcc_data1/cntrlSLOB.ctl
      0 KB         0 KB    read       no      /vx/tpcc_data1/SLOB
    648 KB         0 KB    read       no      /vx/tpcc_data1
```

List SmartIO cache for file system `tpcc_data1`.

```
SmartIO> fs stat tpcc_data1

       Cache Size:   15.97 GB
Cache Utilization:     652 KB ( 0.00 %)

Read Cache
Hit Ratio    Data Read    Data Written

/vx/tpcc_data1:
  96.40 %     12.55 MB        1.656 MB

/vx/tpcc_data1:
READ CACHE    WRITEBACK    MODE       PINNED  NAME
```

16.5.5 SEE ALSO

cache(1), fs(1), file(1)

17 Storage Commands

17.1 storage

17.1.1 SYNOPSIS

`pool create` *pool_name disk1* [, *disk2*,...] [*isolated={yes|no}*]

`pool adddisk` *pool_name disk1* [, *disk2*,...]

`pool destroy` *pool_name*

`pool free` [*pool_name*]

`pool list`

`pool markdiskspare` *pool_name disk1* [, *disk2*,...]

`pool removediskspare` *pool_name disk1* [, *disk2*,...]

`disk configure local` <nodename> <vendor_id> <product_id> [*serial_num*]

`pool mvdisk` *src_pool dest_pool disk1* [, *disk2*,...]

`pool rename` *old_name new_name*

`pool rmdisk` *disk1* [, *disk2*,...]

`disk list [ stats | detail | paths ]`

`disk grow` *disk_name*

`disk format` *disk_name* [, *disk2*,...] [*force*]

`hba` [*host_name*]

`fencing destroy`

`fencing on disk` [*disk1*, *disk2*, *disk3*]

`fencing on majority`

`fencing off`

`fencing replace` *src_disk* fIdest_disk

`fencing status`

`fs create simple` *fs_name size pool1* [, *disk1*,...] [blksize=<bytes>] [pdir_enable={yes|no}] [encrypt={on|off}] [worm={yes|no}]

`fs create mirrored` *fs_name size nmirrors pool1* [, *disk1*,...] [protection=disk|pool] [blksize=<bytes>] [pdir_enable={yes|no}] [encrypt={on|off}] [worm={yes|no}]

`fs create striped` *fs_name size ncolumns pool1* [, *disk1*,...] [blksize=<bytes>] [pdir_enable={yes|no}] [encrypt={on|off}] [worm={yes|no}]

`fs create mirrored-stripe` *fs_name size nmirrors ncolumns pool1* [, *disk1*,...] [protection=disk|pool] [blksize=<bytes>] [pdir_enable={yes|no}] [encrypt={on|off}] [worm={yes|no}]

`fs create striped-mirror` *fs_name size nmirrors ncolumns pool1* [, *disk1*,...] [protection=disk|pool] [blksize=<bytes>] [pdir_enable={yes|no}] [encrypt={on|off}] [worm={yes|no}]

`fs create ecoded` *fs_name size ncolumns nparity pool1* [, *disk1*,...] [blksize=<bytes>] [stripeunit=<kilobytes>] [stripe_aligned={yes|no}] [stripe_tag={node|disk}] [rotating_parity={yes|no}] [workload={virtualmachine|mediaserver}]

`fs create largefs simple` *fs_name size pool1* [*blksize=<bytes>*]

`fs create largefs mirrored` *fs_name size nmirrors pool1* [blksize=<bytes>]

`fs create largefs striped` *fs_name size ncolumns pool1* [stripeunit=<kilobytes>] [blksize=<bytes>]

`fs create largefs mirrored-stripe` *fs_name size nmirrors ncolumns pool1* [stripeunit=<kilobytes>] [blksize=<bytes>]

`fs create largefs striped-mirror` *fs_name size nmirrors ncolumns pool1* [stripeunit=<kilobytes>] [blksize=<bytes>]

`fs create pretuned` *fs_name size pool1* workload={virtualmachine|mediaserver} [layout={striped|striped-mirror|mirrored-stripe}] [ncolumns] [encrypt={on|off}]

`fs defrag now` *fs_name time*

`fs destroy` *fs_name*

`fs list` [*fs_name*]

`fs online` *fs_name*

`fs offline` *fs_name*

`fs growto` primary|secondary *fs_name new_length* [*pool1* [, *disk1*,...]] [protection=disk|pool]

`fs growby` primary|secondary *fs_name length_change* [*pool1* [, *disk1*,...]] [protection=disk|pool]

`fs shrinkto` primary|secondary *fs_name new_length*

`fs shrinkby` primary|secondary *fs_name length_change*

`fs addmirror` *fs_name pool1* [, *disk1*,...][protection=disk|pool]

`fs rmmirror` *fs_name [pool_or_disk_name]*

`fs addcolumn` *fs_name ncolumns pool_or_disk_name*

`fs rmcolumn` *fs_name*

`fs checkmirror`

`fs resync` [*fs_name*]

`fs checkresync`

`fs setfastresync` *fs_name [pool_or_disk_name]*

`fs unsetfastresync` *fs_name*

`fs fsck` *fs_name*

`cloud addservice` *service_name* [service_provider=<service_provider>]

`cloud listservice` [*service_name*]

`cloud removeservice` *service_name*

`tier add simple` *fs_name size pool1* [, *disk1*,...]

`tier add mirrored` *fs_name size nmirrors pool1* [, *disk1*,...] [protection=disk|pool]

`tier add striped` *fs_name size ncolumns pool1* [, *disk1*,...]

`tier add mirrored-stripe` *fs_name size nmirrors ncolumns pool1* [, *disk1*,...] [protection=disk|pool]

`tier add striped-mirror` *fs_name size nmirrors ncolumns pool1* [, *disk1*,...] [protection=disk|pool]

`tier addcolumn` *fs_name ncolumns pool1* [, *disk1*,...]

`tier add cloud` *fs_name tier_name service_name region S3|Glacier*

`tier rmcolumn` *fs_name*

`tier remove` *fs_name* [*tier_name*]

`tier addmirror` *fs_name pool1* [, *disk1*,...][protection=disk|pool]

`tier rmmirror` *fs_name [pool_or_disk_name]*

`tier listfiles` *fs_name* (primary|secondary)

`tier mapfile` *fs_name file_path*

`tier policy list`

`tier policy prune list`

`tier policy prune modify` *fs_name delete_after*

`tier policy modify` *fs_name* (primary|secondary) *days minacctemp period*

`tier relocate` *fs_name dirPath*

`tier policy remove` *fs_name*

`tier policy prune remove` *fs_name*

`tier policy run` *fs_name*

`tier schedule list`

`tier schedule modify` *fs_name minute hour day_of_the_month month day_of_the_week node_name*

`tier schedule remove` *fs_name*

`tier query` *fs_name*

`tier setfastresync` *fs_name pool1* [, *disk1*,...]

`tier unsetfastresync` *fs_name*

`tier stats monitor` *fs_name tier_name* [*interval*]

`tier stats {show|reset|usage}` *fs_name tier_name*

`scanbus`

`snapshot create` *snapshot_name fs_name* [removable=yes|removable=no]

`snapshot restore` *snapshot_name fs_name*

`snapshot destroy` *snapshot_name fs_name*

`snapshot list` [*fs_name*]

`snapshot online` *snapshot_name fs_name*

`snapshot offline` *snapshot_name fs_name*

`snapshot quota on` *fs_name* [capacity_limit]

`snapshot quota off` [*fs_name*]

`snapshot quota list`

`snapshot schedule create` *schedule_name fs_name max_snapshot_limit minute hour day_of_the_month month day_of_the_week*

`snapshot schedule modify` *schedule_name fs_name max_snapshot_limit minute hour day_of_the_month month day_of_the_week*

`snapshot schedule destroyall` *schedule_name fs_name*

`snapshot schedule preserve` *schedule_name fs_name snapshot_name*

`snapshot schedule show` [fs_name] [schedule_name]

`snapshot schedule delete` *fs_name* [schedule_name]

`iscsi status`

`iscsi start`

`iscsi stop`

`iscsi device add` *device*

`iscsi device del` *device*

`iscsi device list`

`iscsi discovery add` *discovery-address*

`iscsi discovery del` *discovery-address*

`iscsi discovery rediscover` *discovery-address*

`iscsi discovery rediscover_new` *discovery-address*

`iscsi discovery list`

`iscsi target add` *target-name portal-address*

`iscsi target del` *target-name* { *discovery-address* | *portal-address* }

`iscsi target login` *target-name* { *discovery-address* | *portal-address* }

`iscsi target logout` *target-name* { *discovery-address* | *portal-address* }

`iscsi target rescan` *target-name*

`iscsi target attr showdefault`

`iscsi target attr showall`

`iscsi target attr show` *target-name*

`iscsi target attr setdefault` *attribute value*

`iscsi target attr setall` *attribute value*

`iscsi target attr set` *target-name attribute value*

`iscsi target list`

`iscsi target listdetail` *target-name*

`iscsi initiatorname set` *initiatorname-prefix*

`iscsi initiatorname list`

`quota fs enable` *fs_name* [`userquota`|`groupquota`]

`quota fs disable` *fs_name* [`userquota`|`groupquota`]

`quota fs status` *fs_name* [`userquota`|`groupquota`]

`quota fs set` `userquota`|`groupquota` *user_or_group_names domain_name* [`hardlimit`|`softlimit`] [`numinodes`|`numspace`] [*value*] [*fs_name*]

`quota fs setall` `userquota` | `groupquota` [`hardlimit` | `softlimit`] [`numinodes` | `numspace`] [*value*] [*fs_name*]

`quota fs setbygroup` *group_names domain_name* [`hardlimit`|`softlimit`] [`numinodes`|`numspace`] [*value*] [*fs_name*]

`quota fs show` *fs_name* [`userquota`|`groupquota`] [*user_or_group_names*] [*domain_name*]

`quota fs setdefault` `userquota`|`groupquota` `hardlimit`|`softlimit` `numinodes`|`numspace` [*value*] [*fs_name*]

`quota fs showdefault` [*fs_name*] [`userquota`|`groupquota`]

`quota cifshomedir enable` [`userquota`|`groupquota`]

`quota cifshomedir disable` [`userquota`|`groupquota`]

`quota cifshomedir status` [`userquota`|`groupquota`]

`quota cifshomedir set` `userquota`|`groupquota` *user_or_group_names domain_name* [`hardlimit` | `softlimit`] [`numinodes`|`numspace`] [*value*]

`quota cifshomedir setall` `userquota` | `groupquota` [`hardlimit` | `softlimit`] [`numinodes` | `numspace`] [*value*]

`quota cifshomedir show` [`userquota`|`groupquota`] [*user_or_group_names*] [*domain_name*]

`quota cifshomedir showdetail` [`userquota`|`groupquota`] [*user_or_group_names*] [*domain_name*]

`quota cifshomedir setdefault` `userquota`|`groupquota` `hardlimit`|`softlimit` `numinodes`|`numspace` [*value*]

`quota cifshomedir showdefault` [`userquota`|`groupquota`]

`dedup enable` *fs_name* [*blksize*]

`dedup disable` *fs_name*

`dedup list` [*fs_name*]

`dedup start` *fs_name* [*node_name*]

`dedup stop` *fs_name*

`dedup status` [*fs_name*]

`dedup set cpu` *cpuvalue* [*fs_name*]

`dedup set memory` *memvalue*

`dedup set priority` *priorityval* [*fs_name*]

`dedup schedule set` *fs_name hours day* [*freq*]

`dedup schedule modify` *fs_name hours day* [*freq*]

`dedup schedule delete` *fs_name*

`dedup dryrun` *fs_name* [*threshold*]

`dedup remove` *fs_name*

`compress file` *fs_name {file|dir} <res_level> [alg_strength] [file_pattern]*

`compress list` *fs_name {file|dir}*

`uncompress file` *fs_name {file|dir} <res_level>*

`compress modage create` *fs_name age_rule*

`compress modage remove` *fs_name*

`compress pattern create` *fs_name pattern_rule*

`compress pattern remove` *fs_name*

`compress schedule create` *sched_name duration minute [hour] [day] [month] [day_of_week] [node]*

`compress schedule remove` *sched_name*

`compress schedule show` *[sched_name]*

`compress schedule start` *fs_name sched_name res_level [alg_strength]*

`compress schedule stop` *fs_name*

`compress schedule list` *fs_name*

`fs defrag schedule create` *sched_name duration minute [hour] [day] [month] [day_of_week] [node]*

`fs defrag schedule remove` *sched_name*

`fs defrag schedule show` *[sched_name]*

`fs defrag schedule start` *fs_name sched_name*

`fs defrag schedule stop` *fs_name*

`fs defrag schedule list` *fs_name*

`fs policy add operation=move` *policy_name fs_name from_tier to_tier retrieval_option=Expedited|Standard|Bulk pattern* [`atime` *condition*] [`mtime` *condition*]

`fs policy add operation=delete` *policy_name fs_name from_tier pattern* [`atime` *condition*] [`mtime` *condition*]

`fs policy modify` *policy_name pattern* [`atime` *condition*] [`mtime` *condition*]

`fs policy delete` *policy_name fs_name*

`fs policy rename` *old_policy_name new_policy_name*

`fs policy list` *[fs_name]*

`fs policy run` *policy_name*

`fs policy dryrun` *policy_name*

`fs policy status` *policy_name*

`fs policy abort` *policy_name*

`fs policy pause` *policy_name*

`fs policy resume` *policy_name*

`fs policy schedule create` *fs_name [minute] [hour] [day_of_the_month] [month] [day_of_the_week]*

`fs policy schedule modify` *fs_name [minute] [hour] [day_of_the_month] [month] [day_of_the_week]*

`fs policy schedule remove` *fs_name*

`fs policy schedule list` *fs_name*

`fs retention set` *path rtime*

`fs retention show` *path*

`fs retention clear` *path*

`fs worm set` *fs_name*

`fs worm clear` *fs_name*

17.1.2 DESCRIPTION

The `storage` commands create logical pools and file systems.

File systems consist of metadata and file data. Metadata consists of information like last modification time, creation time, permissions, and so on. The total amount of space taken by the metadata depends upon the number of files. A file system that contains many smaller files requires more space to store the metadata, and a file system with fewer large files requires less space to store the metadata.

When a file system is created, some space is set aside initially for the metadata. This space is generally proportional to the size of the file system. This is the reason for the non-zero usage percentage in the output of `fs list` just after the creation of the file system. The space that was set aside for the metadata would grow or shrink as and when required. A file system on a 1 GB volume would take around 40 MB (about 4%) initially for storing the metadata, whereas a file system of size 10 MB would take around 7.3 MB (73%) initially for storing the metadata.

17.1.3 OPTIONS

device

Device on which the operation takes place.

target-name

Name of the iSCSI target at which SCSI LUNs are available. *target-name* should conform to the naming rules defined in RFC3721.

encrypt={yes|no}

Specify whether to create the file system on encrypted volume. If set to `on`, file system is created on encrypted volume. The default value is *encrypt=off*.

pdir_enable={yes|no}

Enable or disable partition directory for the file system. The default value is *pdir_enable=no*, which means partition directory is disabled for the file system.

worm={yes|no}

Specify whether to create a worm-enabled file system. If set to `yes`, file system is created with WORM support enabled. The default value is *worm=no*.

workload={virtualmachine|mediaserver}

Type of workload using this file system. The workload type cannot be changed after the file system is created. If the workload type is virtualmachine, then all the files in the file system have a minimum extent size of 1 MB. This reduces file system fragmentation and improves virtual machine I/O performance.

layout={striped|striped-mirror|mirrored-stripe}

Layout of the underlying volume on which the file system is created. The default value is layout=striped.

discovery-address

Discovery address is the target address at which the initiator can request a list of targets using SendTarget's text request as specified in the iSCSI protocol of RFC3720. If no port is specified with the discovery address, the default port 3260 is used. IPv4 and IPv6 addresses are supported. Examples:

```
192.168.0.4
192.168.0.4:3260
2001:c90::211:9ff:feb8:a9e9
[2001:c90::211:9ff:feb8:a9e9]:3260
```

initiatorname-prefix

initiatorname-prefix is a name that conforms to the naming rules for initiator and target names as specified in RFC3721. Initiator names for nodes in the cluster are generated by appending the node number to this prefix.

portal-address

Portal address is the location at which the target is accessible. IPv4 and IPv6 addresses are supported. Examples:

```
192.168.0.4
192.168.0.4,1
192.168.0.4:3260
192.168.0.4:3260,1
2001:c90::211:9ff:feb8:a9e9
2001:c90::211:9ff:feb8:a9e9,1
[2001:c90::211:9ff:feb8:a9e9]:3260
[2001:c90::211:9ff:feb8:a9e9]:3260,10
```

`pool create` *pool_name disk1* [, *disk2*,...] [*isolated={yes|no}*]

Create a logical pool from a given set of disks. If isolated=yes, then a new disk group is created. The default value for isolated is isolated=no.

`pool adddisk` *pool_name disk1* [, *disk2*,...]

Add a set of disks to a logical pool.

`pool destroy` *pool_name*

Destroy a pool.

`pool free` [*pool_name*]

List free space in each of the pools.

`pool list`

List all the pools.

`pool mvdisk` *src_pool dest_pool disk1* [, *disk2*,...]

Move disks from one pool to another.

`pool rename` *old_name new_name*

Rename a pool.

`pool rmdisk` *disk1* [, *disk2*,...]

Remove disks from a pool.

`pool markdiskspare pool_name` *disk1* [, *disk2*,...]

Mark a disk as a spare disk and add it to a pool which is later used for hot-relocation. In case of failure of a disk or a plex, the affected subdisks are relocated to disks designated as spare disks.

`pool removediskspare pool_name` *disk1* [, *disk2*,...]

Remove the spare disk flag set on a disk by `markdiskspare` command.

`disk list stats`

Display the aggregated information of all the disks.

`disk list detail`

Display properties of all the disks.

`disk list paths`

Display the list of multiple paths of all the disks from all the nodes in the cluster.

`disk configure local` <nodename> <vendor_id> <product_id> [serial_num]

Adds local disks having the specified vendor ID and product ID in the JBOD category.

`disk grow` *disk_name*

Increase the size of the specified disk. Before performing this operation make sure that you increase the storage capacity of the disk on the storage array. `Caution:` When increasing the storage capacity of a disk, make sure that the storage array does not reformat it. This will destroy the data. For help, contact your Storage Administrator.

`disk format` *disk1* [, *disk2*,...] [`force`]

If the disk does not belong to any disk group, the format option erases the first 100 M space on the disk, and the disk is formatted using the `vxdisksetup` command. Use the `force` option to force formatting of disks with invalid configurations.

`hba` [*host_name*]

Display World Wide Name (WWN) information of all or a specified node of the cluster.

`fencing destroy`

Destroy coordinator pool. This command has no effect for majority-based fencing.

`fencing on disk` [*disk1* , *disk2* , *disk3*]

Enable I/O fencing on the three disks (only three disks required).

`fencing on majority`

Enable majority-based I/O fencing.

`fencing off`

Disable I/O fencing on all the nodes.

`fencing replace` *src_disk dest_disk*

Replace an existing coordinator disk with another disk. If the disk being replaced is in a failed state, then it is mandatory to delete the disk from the array. This is required because if the failed disk comes up and works properly, it can lead to an even number of I/O fencing disks, and this affects the functionality. This command has no effect for majority-based I/O fencing.

`fencing status`

Check I/O fencing status.

`fs create simple` *fs_name size pool1* [, *disk1*,...] [blksize=<bytes>] [pdir_enable={yes|no}] [encrypt={on|off}] [worm={yes|no}]

Create a simple file system with a specified size on one of the specified pools/disks and make it online.

`fs create mirrored` *fs_name size nmirrors pool1* [, *disk1*,...] [protection=disk|pool] [blksize=<bytes>] [pdir_enable={yes|no}] [encrypt={on|off}] [worm={yes|no}]

Create a mirrored file system with a specified number of mirrors and make it online.

`fs create striped` *fs_name size ncolumns pool1* [, *disk1*,...] [blksize=<bytes>] [pdir_enable={yes|no}] [encrypt={on|off}] [worm={yes|no}]

Create a striped file system with a specified number of stripes and make it online.

`fs create mirrored-stripe` *fs_name size nmirrors ncolumns pool1* [, *disk1*,...] [protection=disk|pool] [blksize=<bytes>] [pdir_enable={yes|no}] [encrypt={on|off}] [worm={yes|no}]

Create a mirrored-striped file system with a specified number of mirrors and stripes and make it online.

`fs create striped-mirror` *fs_name size nmirrors ncolumns pool1* [, *disk1*,...] [protection=disk|pool] [blksize=<bytes>] [pdir_enable={yes|no}] [encrypt={on|off}] [worm={yes|no}]

Create a striped-mirrored file system with a specified number of mirrors and stripes and make it online.

`fs create largefs simple` *fs_name size pool1* [blksize=<bytes>]

Create a largefs file system with the specified size on pools/disks specified and bring it online.

`fs create largefs mirrored` *fs_name size nmirrors pool1* [blksize=<bytes>]

Create a mirrored largefs file system with the specified number of mirrors and make it online.

`fs create largefs striped` *fs_name size ncolumns pool1* [stripeunit=<kilobytes>] [blksize=<bytes>]

Create a striped largefs file system with the specified number of stripes and make it online.

`fs create largefs mirrored-stripe` *fs_name size nmirrors ncolumns pool1* [stripeunit=<kilobytes>] [blksize=<bytes>]

Create a mirrored-striped largefs file system with a specified number of mirrors and stripes and make it online.

`fs create largefs striped-mirror` *fs_name size nmirrors ncolumns pool1* [stripeunit=<kilobytes>] [blksize=<bytes>]

Create a striped-mirrored largefs file system with a specified number of mirrors and stripes and make it online.

`fs create ecoded` *fs_name size ncolumns nparity pool1* [, *disk1*,...] [blksize=<bytes>] [stripeunit=<kilobytes>] [stripe_aligned={yes|no}] [stripe_tag={node|disk}] [rotating_parity={yes|no}] [workload={virtualmachine|mediaserver}]

Create an ecoded file system with a specified number of data columns and parity columns and make it online.

`fs create pretuned` *fs_name size pool1* workload={virtualmachine|mediaserver} [layout={striped|striped-mirror|mirrored-stripe}] [ncolumns] [encrypt={on|off}]

Create a file system with the specified size using the disks from the specified pool. The volume is created with the specified layout. Stripe the volume across the specified number number of columns with 512 K as the stripe size. Tune the file system for the specified workload.

`fs defrag now` *fs_name time*

Defragment a file system now. The *time* value should be larger than 1 minute or infinite.

`fs destroy` *fs_name*

Destroy a file system.

`fs list` [*fs_name*]

List all the file systems.

`fs online` *fs_name*

Make the file system online.

`fs offline` *fs_name*

Make the file system offline.

`fs growto` primary|secondary *fs_name new_length* [*pool1* [, *disk1*,...]] [protection=disk|pool]

Grow the file system's primary or secondary tier to a specified size. Or grow the largefs file system's primary disk tier to a specified size.

`fs growby` primary|secondary *fs_name length_change* [*pool1* [, *disk1*,...]] [protection=disk|pool]

Grow the file system's primary or secondary tier by a specified size. Or grow the largefs file system's primary disk tier by a specified size.

`fs shrinkto` primary|secondary *fs_name new_length*

Shrink the file system's primary or secondary tier to a specified size. Or shrink the largefs file system's primary disk tier to a specified size.

`fs shrinkby` primary|secondary *fs_name length_change*

Shrink the file system's primary or secondary tier by a specified size. Or shrink the largefs file system's primary disk tier by a specified size.

`fs addmirror` *fs_name pool1* [, *disk1*,...] [protection=disk|pool]

Add a mirror to the file system.

`fs rmmirror` *fs_name [pool_or_disk_name]*

Remove a mirror from the file system that is spanning on the specified pools and disks.

`fs addcolumn` *fs_name ncolumns pool_or_disk_name*

Add the specified number of columns to the file system. In case of a striped file system, the number of disks specified should be equal to *ncolumns*. In case of mirrored-stripe and striped-mirror, the disks specified should be equal to (ncolumns * number_of_mirrors_in_fs)

`fs rmcolumn` *fs_name*

Remove a column from the file system.

`fs checkmirror`

Show file systems that have stale mirrors.

`fs resync` [*fs_name*]

Resynchronize all stale mirrors for all file systems or for a certain file system.

`fs checkresync`

Show resynchronization progress running in the background.

`fs setfastresync` *fs_name [pool_or_disk_name]*

Enable fast resync for the specified file system.

`fs unsetfastresync` *fs_name*

Disable fast resync for the specified file system.

`fs fsck` *fs_name*

Check and repair the specified file system.

`cloud addservice` *service_name* [service_provider=<service_provider>]

Add a public cloud subscription to Veritas Access as a cloud service with a name. The added subscription is accessed using *service_name* thereafter. You are prompted to provide the subscription credentials. Credentials are verified and the cloud service is added.

`cloud listservice` [*service_name*]

List added cloud services. If *service_name* is provided, then only that cloud service is listed. Otherwise all configured services are listed.

`cloud removeservice` *service_name*

Remove the cloud service *service_name*.

Warning: Service removal fails if there are active cloud tiers associated with this service at time of removal.

`tier add simple` *fs_name size pool1* [, *disk1*,...]

Add a simple tier to the specified file system on one of the specified pools and disks.

`tier add mirrored` *fs_name size nmirrors pool1* [, *disk1*,...] [protection=disk|pool]

Add a mirrored tier to the specified file system.

`tier add striped` *fs_name size ncolumns pool1* [, *disk1*,...]

Add a striped tier to the specified file system.

`tier add mirrored-stripe` *fs_name size nmirrors ncolumns pool1* [, *disk1*,...] [protection=disk|pool]

Add a mirrored-stripe tier to the specified file system.

`tier add striped-mirror` *fs_name size nmirrors ncolumns pool1* [, *disk1*,...] [protection=disk|pool]

Add a striped-mirror tier to the specified file system.

`tier addcolumn` *fs_name ncolumns pool1* [, *disk1*,...]

Add a column to the tier of the file system.

`tier add cloud` *fs_name tier_name service_name region S3|Glacier*

Add a cloud tier to the specified scale-out file system using service name and region. You can specify if you want to add an S3-compatible or Glacier cloud storage tier by specifying the appropriate option. S3-compatible is any third-party implementation of the Amazon S3 APIs (Supporting Signature version 4 signing).

The supported regions of the Amazon S3 service are the following:

- California (us-west-1)
- Central (ca-central-1)
- Frankfurt (eu-central-1)
- Ireland (eu-west-1)
- London(eu-west-2)
- Mumbai (ap-south-1)
- Ohio (us-east-2)
- Oregon (us-west-2)
- Seoul (ap-northeast-2)
- Singapore (ap-southeast-1)
- Sydney (ap-southeast-2)
- SaoPaulo (sa-east-1)
- Tokyo (ap-northeast-1)
- Virginia (us-east-1)

Region is irrelevant to an S3-compatible cloud storage service. If you select an S3-compatible service, select "none" as the region. Any other region specified with an S3-compatible is ignored.

Warning: When a cloud storage service is used as a cloud tier for a scale-out file system, Veritas Access exclusively owns all the buckets and the objects created by Veritas Access. Any attempt to tamper with these buckets or objects outside of Veritas Access corrupts the files represented by those modified objects.

`tier rmcolumn` *fs_name*

Remove a column from the tier of the file system.

`tier remove` *fs_name* [*tier_name*]

Remove a tier from the file system. All files on the secondary tier get relocated to the primary tier. Ensure that you remove the policy by running `tier policy remove <fs_name>` before running the tier remove command.

`tier addmirror` *fs_name pool1* [, *disk1*,...] [protection=disk|pool]

Add a mirror to the tier of the file system.

`tier rmmirror` *fs_name [pool_or_disk_name]*

Remove a mirror from the tier of the file system that is spanning on the specified pools and disks.

`tier listfiles` *fs_name* (primary|secondary)

Show all the files that are on the specified tier.

`tier mapfile` *fs_name file_path*

Show the tier location of the specified file. The path of the file is relative to the file system. For example, `tier mapfile fs1 /a.txt` would show the location of `a.txt`, which is in the root directory of the file system `fs1`.

`tier policy list`

Show policy for each tiered file system.

`tier policy prune list`

Show prune policy for each tiered file system.

`tier policy prune modify` *fs_name delete_after*

Change the prune policy of the tiered file system.

`tier policy modify` *fs_name* (primary|secondary) *days minacctemp period*

Change the policy of the tiered file system.

`tier relocate` *fs_name dirPath*

Relocate the directory or the file from the secondary tier to the primary tier. This does not relocate NDS (Named Data Stream also includes extended attributes) to the primary tier.

`tier policy remove` *fs_name*

Remove the policy of the tiered file system.

`tier policy prune remove` *fs_name*

Remove the prune policy of the tiered file system.

`tier policy run` *fs_name*

Run the policy of the tiered file system.

`tier schedule list`

Show schedules of all the tiered file systems.

`tier schedule modify` *fs_name minute hour day_of_the_month month day_of_the_week node_name*

Modify the schedule of the tiered file system.

`tier schedule remove` *fs_name*

Remove the schedule of the tiered file system.

`tier query` *fs_name*

Show the list of files that will be moved by running the policy.

`tier setfastresync` *fs_name pool1* [, *disk1*,...]

Set fastresync for the tier of the file system.

`tier unsetfastresync` *fs_name*

Unset fastresync for the tier of the file system.

`tier stats {show|reset|usage}` *fs_name tier_name*

Show the access statistics, reset the access statistics to zero, or show the total usage of the specified cloud tier in the largefs file system.

`tier stats monitor` *fs_name tier_name* [*interval*]

Monitor the access statistics of the specified cloud tier in the largefs file system. The default interval is five seconds.

`scanbus`

Scan all the SCSI devices connected to all or a specified node of the cluster.

`snapshot create` *snapshot_name fs_name* [removable=yes|removable=no]

Create a snapshot for a specified file system. If the removable attribute is `yes` and if it is offline, then it is removed automatically, if the file system runs out of space.

`snapshot restore` *snapshot_name fs_name*

Restore a file system by a given snapshot.

`snapshot destroy` *snapshot_name fs_name*

Destroy the snapshot of a file system.

`snapshot list` [*fs_name*]

Displays all the snapshots of the specified file system. If no file system is specified, snapshots of all the file systems are displayed.

`snapshot online` *snapshot_name fs_name*

Make the snapshot online.

`snapshot offline` *snapshot_name fs_name*

Make the snapshot offline.

`snapshot quota on` *fs_name* [capacity_limit]

Disallow creation of snapshots on the given file system when the space used by all the snapshots of that file system exceeds a given capacity limit. The space used by the snapshots is not restricted.

`snapshot quota off` [*fs_name*]

Disable quota capacity limit for a specified system.

`snapshot quota list`

Display snapshot quota information of all the file systems.

`snapshot schedule create` *schedule_name fs_name max_snapshot_limit minute hour day_of_the_month month year day_of_the_week*

Create schedule for automated snapshot creation of a particular file system.

`snapshot schedule modify` *schedule_name fs_name max_snapshot_limit minute hour day_of_the_month month year day_of_the_week*

Modify schedule for automated snapshot creation of a particular file system.

`snapshot schedule destroyall` *schedule_name fs_name*

Destroy all automated snapshots corresponding to a given schedule name and file system name (excluding the preserved and online snapshots).

`snapshot schedule preserve` *schedule_name fs_name snapshot_name*

Preserve a limited number of snapshots corresponding to an existing schedule and a specific file system name so that they are not removed as part of the snapshot schedule autoremove command.

`snapshot schedule show` [fs_name] [schedule_name]

Show all schedules that have been set for automated snapshot creation.

`snapshot schedule delete` *fs_name* [schedule_name]

Deletes the schedule set for automated snapshot creation for a particular file system or for a particular schedule.

`iscsi status`

Shows the status of the iSCSI initiator service.

`iscsi start`

Starts the iSCSI initiator service.

`iscsi stop`

Stops the iSCSI initiator service.

`iscsi device add` *device*

Add a device for use with the iSCSI initiator. iSCSI initiator connections use this device to connect to the target. If there are any existing targets, then the iSCSI initiator initiates a connection to all the targets using *device*.

`iscsi device del` *device*

Delete a *device* from the iSCSI initiator configuration. Any existing connections from *device* to targets is terminated. If *device* is the last device in the iSCSI initiator configuration and there are existing targets, then the device cannot be deleted from the configuration.

`iscsi device list`

List the devices used by the iSCSI initiator.

`iscsi discovery add` *discovery-address*

Add a discovery address to the iSCSI initiator configuration. If no TCP port is specified with the *discovery-address*, then the default port 3260 is used. Any targets discovered at *discovery-address* are automatically logged on to.

`iscsi discovery del` *discovery-address*

Delete a discovery address from an iSCSI initiator configuration. Any targets discovered using *discovery-address* are also deleted from the configuration.

`iscsi discovery rediscover` *discovery-address*

Perform a discovery of changes in targets or LUNs at *discovery-address*. Any LUNs or targets that have been removed at *discovery-address* are automatically removed from the configuration. New LUNs or targets discovered at *discovery-address* are automatically added and logged on to.

`iscsi discovery rediscover_new` *discovery-address*

Perform a discovery of changes in targets or LUNs at *discovery-address*. New LUNs or targets discovered at *discovery-address* are automatically added and logged on to. Does not discover any targets that have been deleted at fIdiscovery- addressfP.

`iscsi discovery list`

List the discovery addresses present in the iSCSI initiator configuration.

`iscsi target add` *target-name portal-address*

Add a static target-portal combination to the iSCSI initiator configuration. The *portal-address* cannot be the same as any *discovery-address* present in the iSCSI initiator configuration. Connections to *portal-address* are made for *target-name*, but no discovery is done for any other targets available at *portal-address*. If no portal tag is specified with *portal-address*, the default portal tag of 1 is used.

`iscsi target del` *target-name* { *discovery-address* | *portal-address* }

Delete a target *target-name* from the iSCSI initiator configuration. Any existing connections to *target-name* are terminated. *discovery-address* or *portal-address* is the address through which the target became visible to the initiator. A target that was discovered at a *discovery-address* once deleted from the iSCSI initiator configuration would again be visible to the iSCSI initiator if re-discovery is done either through `iscsi discovery rediscover` or `scanbus` commands.

`iscsi target login` *target-name* { *discovery-address* | *portal-address* }

Logon to a target *target-name* from the iSCSI initiator. Connections to *target-name* are made from all devices present in the iSCSI initiator configuration. *discovery-address* or *portal-address* is the address through which the target became visible to the initiator.

`iscsi target logout` *target-name* { *discovery-address* | *portal-address* }

Logout from connections to *target-name* from the iSCSI initiator. *discovery-address* or *portal-address* is the address through which the target became visible to the initiator. A target once logged out by the iSCSI initiator will not be logged in to until `iscsi target login` is requested.

`iscsi target rescan` *target-name*

Rescan target *target-name* for new LUNs.

`iscsi target attr showdefault`

Show the default value for target attributes.

`iscsi target attr showall`

Show the value of target attributes for all the known targets.

`iscsi target attr show` *target-name*

Show the value of target attributes for target *target-name*

`iscsi target attr setdefault` *attribute value*

Set the default value of *attribute* to *value*. Default value is inherited by any new targets that get added.

`iscsi target attr setall` *attribute value*

Set the value of *attribute* to *value* for all the known targets. This does not change the default value as shown in `iscsi target attr showdefault`. Changes to the values are effective after re-login.

`iscsi target attr set` *target-name attribute value*

Set the value of *attribute* to *value* for *target-name*. Changes to the values are effective after re-login.

`iscsi target list`

List the targets visible to the iSCSI initiator. A target can be in any one of ONLINE, OFFLINE, or RETRY states.

`iscsi target listdetail` *target-name*

List detailed information about the target *target-name*.

`iscsi initiatorname setprefix` *initiatorname-prefix*

Set the prefix used to generate initiator names. Initiator names are generated as *initiatorname-prefix* followed by the node number of the node.

`iscsi initiatorname list`

List the initiator names for all the nodes in the cluster.

`quota fs enable` *fs_name* [`userquota` | `groupquota`]

`quota cifshomedir enable` [`userquota` | `groupquota`]

Enable user or group quota on a file system or CIFS home directories.

`quota fs disable` *fs_name* [`userquota` | `groupquota`]

`quota cifshomedir disable` [`userquota` | `groupquota`]

Disable user or group quota on a file system or CIFS home directories.

`quota fs status` *fs_name* [`userquota`|`groupquota`]

`quota cifshomedir status` [`userquota`|`groupquota`]

Show the status of quota settings on a file system or CIFS home directories. Quota setting status only shows if the quota is enabled or disabled.

`quota fs set userquota | groupquota` *user_or_group_names domain_name* [`hardlimit``|``softlimit`] [`numinodes`|`numspace`] [*value*] [*fs_name*]

`quota cifshomedir set userquota | groupquota` *user_or_group_names domain_name* [`hardlimit``|``softlimit`] [`numinodes`|`numspace`] [*value*]

Set quota value for the users or groups on a file system or CIFS home directories. If *value* is not specified, then the default value set from the respective `setdefault` commands is used to configure the quota limit. If *value* is 0, it is treated as unlimited quota. If all values of user or group quota are 0, the user or group are automatically deleted from the quota settings, which means `quota fs status` does not show this user's or group's settings as all quota values are unlimited for it.

`quota fs setall userquota | groupquota [hardlimit | softlimit] [numinodes | numspace]` [*value*] [*fs_name*]

`quota cifshomedir setall userquota | groupquota [hardlimit | softlimit] [numinodes | numspace]` [*value*]

Set quota value for all the users and groups for whom the quota has already been set with `set` commands. Other users and groups (for whom quota has not been set previously) are not affected. If *value* is not specified, then the default value set from the respective `setdefault` commands is used to configure the quota limit. If *value* is 0, it is treated as an unlimited quota. If all values of user or group quota are 0, the user or group are automatically deleted from the quota settings, which means `quota fs status` does not show this user's or group's settings, as all quota values are unlimited for it.

`quota fs setbygroup` *group_names domain_name* [`hardlimit`|`softlimit`] [`numinodes`|`numspace`] [*value*] [*fs_name*]

Set the user quota for users of specified groups.

`quota fs show` *fs_name* [`userquota`|`groupquota`] [*user_or_group_names*] [*domain_name*]

`quota cifshomedir show` [`userquota`|`groupquota`] [*user_or_group_names*] [*domain_name*]

`quota cifshomedir showdetail` [`userquota`|`groupquota`] [*user_or_group_names*] [*domain_name*]

Show the quota values that are already set. This also shows the consumed (used space) quota of the users and groups. `cifshomedir show` shows the general quota values on the CIFS home directories. `cifshomedir showdetail` shows the detailed quota values set on each file system for CIFS home directories.

`quota fs setdefault userquota``|``groupquota hardlimit | softlimit numinodes | numspace` [*value*] [*fs_name*]

`quota cifshomedir setdefault userquota|groupquota hardlimit|softlimit numinodes|numspace` [*value*]

Set the default value that is used for the quota limits. The values are stored in the configuration file only. The actual application of quota can be done with `set` and `setall` commands using these default values.

`quota fs showdefault` [*fs_name*] [`userquota`|`groupquota`]

`quota fs showdefault` [`userquota`|`groupquota`]

Show the default quota values from the configuration file.

`dedup enable` *fs_name blksize*

Enable deduplication on a file system. Deduplication must be enabled for a file system before setting its configuration parameters and schedule. This command also re-enables a deduplication schedule for a file system. *blksize* should be specified in bytes e.g. 4096. It should be the power of 2, multiple of file system's block size and equal to or less than 128 K. Default block size is 0, which means that if file system's block size is less than 4096 bytes, then the dedup block size is set to 4096 bytes. Otherwise the dedup block size is set the same as the file system's block size.

`dedup disable` *fs_name*

Disable deduplication schedule on a file system. Other information like configuration, schedule, and deduplication database remains intact. **Warning:** keeping deduplication disabled for longer periods of time may reduce the effectiveness of deduplication when it is re-enabled.

`dedup list` [*fs_name*]

List the deduplication configured file systems along with its granularity, state, schedule, and node list where it runs. Currently, node list is picked up automatically by the system and cannot be set by the user.

`dedup start` *fs_name* [*node_name*]

Start deduplication manually. *fs_name* must be enabled for deduplication to start. You can specify the node to run the deduplication.

`dedup stop` *fs_name*

Stop the deduplication process running on a file system.

`dedup status` [*fs_name*]

Display the current status of deduplication. It shows the file system name, saving amount, status, node where deduplication is either running or has completed, deduplication type i.e., SCHEDULED or MANUAL and other details.

`dedup set cpu` *cpuvalue* [*fs_name*]

Set the CPU limit. This parameter can be set as global and/or local. Possible *cpuval* are *idle* and *yield*.

`dedup set memory` *memvalue*

Set the memory limit in MB. It should be in range 128-4096 MB, default value is 128 MB. This can be set only as a global parameter.

`dedup set priority` *priorityval* [*fs_name*]

Set priority for deduplication-enabled file system(s). This parameter can be set as global and/or local. Possible *priorityval* are *low*, *normal*, and *high*.

`dedup schedule set` *fs_name hours day* [*freq*]

Set schedule for deduplication-enabled file system. This can be set only as a local parameter. Two categories of schedule are allowed i.e., run periodicity and type periodicity. The granularity of schedule is limited to time of day and day of month. *hours* can be specified as * (every hour), */N (every N hours) where N is in range [1-12] or 5 comma-separated hours list in the range [0-23]. *day* can be specified as * (every day), */N (every N days), where N is in range [1-15] or any number in the range [1-7], where 1 means Sunday, 2 means Monday, and so on. *freq* can be specified in the range 1-5. Default value is 1. A value of 4 means every 4th run deduplicates the file system, whereas the other runs consolidate the changes.

`dedup schedule modify` *fs_name hours day* [*freq*]

Modify the deduplication schedule set above. *hours*, *day*, and *freq* values can be specified same as above mentioned in *dedup schedule set* command.

`dedup schedule delete` *fs_name*

Delete the deduplication schedule.

`dedup dryrun` *fs_name* [*threshold*]

Initiate a deduplication dry run on the file system *fs_name*. This reports the approximate saving of space. Dry run can be converted to the actual deduplication if it meets the threshold value.

`dedup remove` *fs_name*

Remove deduplication-configuration related information from the specified file systems *fs_name*. This command removes all configurations and the database specific to that file system. `Warning:` This operation cannot be undone and re-running deduplication may take significant time.

`compress file` *fs_name* fI{file|dir} <*res_level*> *[alg_strength] [file_pattern]*

Compress named file or directory using specified <*res_level*> and *alg_strength*. The default alg-strength is 6. If a directory is specified, then all the files under the named directory are compressed. For directories, you can specify file name patterns. The <*res_level*> can be low, medium, or high.

`compress list` *fs_name {file|dir}*

List compression details for the named file or directory.

`uncompress file` *fs_name* fI{file|dir} <*res_level*>

Uncompress the named file or directory using specified <*res_level*>.

`compress modage create` *fs_name age_rule*

Add an age-based rule for scheduled compression jobs for the file system with name *fs_name*. The *age_rule* is specified in days.

`compress modage remove` *fs_name*

Remove the age-based rule for the *fs_name* file system.

`compress pattern create` *fs_name pattern_rule*

Add a pattern-based rule for scheduled-compression jobs for the file system with name *fs_name*.

`compress pattern remove` *fs_name*

Remove the pattern-based rule for the *fs_name* file system.

`compress schedule create` *sched_name duration minute [hour] [day] [month] [day_of_week] [node]*

Create a compression schedule by name *sched_name*. The time and frequecy of this schedule are specified using the command line in crontab format. This scheduled job runs on the node specified by *[node]*. If this node is not online at the time of the job, then this particular compression job runs on the CFS primary.

`compress schedule remove` *sched_name*

Remove the compression schedule by name *sched_name*. Make sure none of the file systems have this *sched_name* assigned.

`compress schedule show` *[sched_name]*

Show the compression schedule information for the schedule by name *sched_name*. If *sched_name* is not specified, then information about all the schedules are displayed.

`compress schedule start` *fs_name sched_name res_level [alg_strength]*

Assign or start the compression schedule by name *sched_name* for the file system by name *fs_name*. The compression job uses resource and algorithm strength specified in args.

`compress schedule stop` *fs_name*

Stop the scheduled compression schedule for the file system by name *fs_name*

`compress schedule list` *fs_name*

List the scheduled compression job details for the file system by name *fs_name*.

`fs defrag schedule create` *sched_name duration minute [hour] [day] [month] [day_of_week] [node]*

Creates a defrag schedule. The time and frequency of this schedule are specified in the command line in crontab format. This scheduled job runs on the node specified. If this node is not online at the time of the job, then this particular defrag job runs on the CFS primary. The scheduled defrag job may last up to the specified duration hours or minutes.

`fs defrag schedule remove` *sched_name*

Remove the defrag schedule by name *sched_name*. Make sure none of the file systems have this *sched_name* assigned.

`fs defrag schedule show` *[sched_name]*

Show the defrag schedule information for the schedule by name *sched_name*. If *sched_name* is not specified, then information about all the schedules are displayed.

`fs defrag schedule start` *fs_name sched_name*

Assign or start the defrag schedule by name *sched_name* for the file system by name *fs_name*.

`fs defrag schedule stop` *fs_name*

Stop the scheduled defrag schedule for the file system by name *fs_name*.

`fs defrag schedule list` *fs_name*

List the scheduled defrag job details for the file system by name *fs_name*.

`fs policy add operation=move` *policy_name fs_name from_tier to_tier retrieval_option=Expedited\Standard\Bulk pattern* [`atime` *condition*] [`mtime` *condition*]

Create a data movement policy for the scale-out file system *fs_name*. The policy moves files and directories that meet the criteria from the source tier (*from_tier*) to the destination tier (*to_tier*). Movement can be from the cloud tier to a disk tier or from a disk tier to the cloud tier. The retrieval option determines the time needed to move files from Amazon Glacier to on-premises. This option is not used when moving files from on-premises to Amazon Glacier. Expedited retrievals typically complete within 1-5 minutes. The expedited option is expensive and you should use it conservatively. Files moved from the Amazon Glacier tier with the expedited option might return with the following error: InsufficientCapacityException (503 service unavailable). This error occurs if there is insufficient capacity to process the expedited request.

This error only applies to expedited retrievals and not to standard or bulk retrievals. Standard retrievals typically complete within 3-5 hours. Bulk retrievals typically complete within 5-12 hours. Bulk is the default option. **Note:** The maximum file size for moving files to AWS Glacier is 4 GB. Pattern identifies the files or directories that you want to move between tiers. Pattern is required. You can further restrict the files or directories to move by specifying the last accessed time (`atime`) or the last modified time (`mtime`). The `atime` and `mtime` criteria are optional.

`fs policy add operation=delete` *policy_name fs_name from_tier pattern* [`atime` *condition*] [`mtime` *condition*]

Create a data deletion policy for the scale-out file system *fs_name*. The policy deletes files and directories that meet the criteria from the specified tier (*from_tier*). Pattern identifies the files or directories that you want to delete. Pattern is required. You can further restrict the files or directories to delete by specifying the last accessed time (`atime`) or the last modified time (`mtime`). The `atime` and `mtime` criteria are optional.

`fs policy modify` *policy_name pattern* [`atime` *condition*] [`mtime` *condition*]

Modify the pattern, atime, and mtime search criteria for the file movement or deletion of the policy. atime and mtime are optional criteria. You cannot change the policy operation or change the storage tiers that were specified when the policy was created.

`fs policy delete` *policy_name fs_name*

Stop any data movement or data deletion policy that was set for a file system. You cannot delete a policy if the policy is running.

`fs policy rename` *old_policy_name new_policy_name*

Rename an existing policy to a new policy name. You cannot rename a policy if the policy is running.

`fs policy list [fs_name]`

List all the data movement and data deletion polices that are set for all the file systems. If *fs_name* is included in the command, then the command lists all the policies for the specified file system.

`fs policy run` *policy_name*

Move or prune files according to the configured file system policy *policy_name*. The policy runs in the background until it gets completed, or until you abort or pause the policy. Filesystem policies are configured only for scale-out file systems (largefs type). If a policy run encounters an error, it is retried 5 times before the run is aborted.If the policy is already active but paused, you are prompted on whether you want to resume the paused job or you want to start a new run. You cannot run a policy if poilcy is scheduled or the last instance is still running. You also cannot run a policy if a tier move command is running for the same path or file system.

`fs policy dryrun` *policy_name*

Perform a dry run of the file system policy *policy_name*. This command previews the running of the policy, but does not actually move or prune any files. Use this command to estimate the I/O activity involved if the policy is run. The dry run collects the statistics, such as the number of files and the amount of data that will be moved. You cannot perform a dry run if the policy is already running, or if a `tier move` command is running on the same path or file system.

`fs policy status` *policy_name*

Show the status of the currently running policy or dry run of the policy *policy_name*. If the policy is not currently running, the status shows the most recent run. The information includes the type of the run (normal or dry run), the status, the total data, the amount of data moved so far, the number of files, the number of files moved or deleted so far, and the name of the last file completed by the policy run.

`fs policy abort` *policy_name*

Abort the currently executing policy run or dry run for the policy *policy_name*. The abort is immediate and interrupts any data movement in progress. The aborted policy cannot be resumed using the `fs policy resume` command. If you need to restart the policy, start a new run with the `fs policy run` command.

`fs policy pause` *policy_name*

Pause the currently executing policy run or dry run for the policy *policy_name*. The pause is immediate and interrupts any data movement in progress. You can resume the policy using the `fs policy resume` command. You can abort the paused policy using the `fs policy abort` command. When you pause a policy, the scheduled runs of the policy are also skipped until the policy is resumed.

`fs policy resume` *policy_name*

Restarts the policy run or dry run that was paused with the `fs policy pause` command. The policy resumes from the point where the policy run was interrupted.

`fs policy schedule create` *fs_name [minute] [hour] [day_of_the_month] [month] [day_of_the_week]*

Create the file system policy schedule for the file system *fs_name*. The schedule uses a time format similar to the format used in UNIX cron configuration files. The schedule applies to the file system policies created using the `fs policy create` command. When a schedule is set for a particular file system, all the policies for that file system are started at the scheduled times. If any policy is paused using the `fs policy pause` command, the policy does not run at the scheduled time. The policy is skipped until the policy is resumed using the `fs policy resume` command.

About the schedule format:

A schedule is specified in a format similar to the UNIX crontab format. The format uses five fields to specify when the schedule runs:

`minute` Enter a numeric value between 0-59, or an asterisk (*), which represents every minute. You can also enter a step value (*/x), or a range of numbers separated by a hyphen.

`hour` Enter a numeric value between 0-23, or an asterisk (*), which represents every hour. You can also enter a step value (*/x), or a range of numbers separated by a hyphen.

`day_of_the_month` Enter a numeric value between 1-31, or an asterisk (*), which represents every day of the month. You can also enter a step value (*/x), or a range of numbers separated by a hyphen.

`month` Enter a numeric value between 1-12, or an asterisk (*), which represents every month. You can also use the names of the month. Enter the first three letters of the month (you must use lowercase letters). You can also enter a step value (*/x), or a range.

`day_of_the_week` Enter a numeric value between 0-6, where 0 represents Sunday, or an asterisk (*), which represents every day of the week.

You can also enter the first three letters of the week (you must use lowercase letters). You can also enter a step value (*/x), or a range.

A step value (*/x) specifies that the schedule runs at an interval of x. The interval should be an even multiple of the field's range. For example, you could specify */4 for the hour field to specify every four hours, since 24 is evenly divisible by 4. However, if you specify */15, you may get undesired results, since 24 is not evenly divisible by 15. The schedule would run after 15 hours, then 7 hours.

A range of numbers (two values separated by a hyphen) represents a time period during which you want the schedule to run.

Examples:

- To run the schedule every two hours every day: 0 */2 * * *
- To run the schedule on 2:00 a.m. every Monday: * 2 * * 1
- To run the schedule at 11:15 p.m. every Saturday: 15 23 * * 6

fs policy schedule modify fs_name [minute] [hour] [day_of_the_month] [month] [day_of_the_week]
Modifies the file system policy schedule for the file system *fs_name*. The schedule uses a time format similar to the format used in UNIX cron configuration files. Refer to the `fs policy schedule create` section for the detailed description of the schedule format.

`fs policy schedule remove` ***fs_name*** Removes the file system policy schedule associated with the file system *fs_name*.

`fs policy schedule list` ***fs_name*** Lists the file system policy schedule associated with the file system *fs_name*.

`fs retention set` ***path rtime*** Sets the retention on a file path *path* or on all the files that are currently present in specified path *path*. *rtime* can be in [1-9](d|D|m|M|y|Y) or mm-dd-yyyy format.

fs retention show ***path*** Shows the retention value applied on the specified file path *path*.

fs retention clear ***path*** Clears the retention on a file path *path* or on all the files that are currently present in specified path *path*.

fs worm set ***fs_name*** Enable worm support for the specified file system.

fs worm clear ***fs_name*** Disable worm support for the specified file system.

17.1.4 EXAMPLES

Display list of disks in tabular form.

```
Storage> disk list stats
Disk             nasgw9_2  nasgw9_1  nasgw9_3  nasgw9_4  nasgw9_5  nasgw9_6
====             ======    ======    ======    ======    ======    ======
AMS_WMS0_0       OK        OK        OK        OK        OK        OK
AMS_WMS0_1       OK        OK        OK        OK        OK        OK
DS4800-0_0       OK        OK        OK        OK        OK        OK
DS4800-0_1       OK        OK        OK        OK        OK        OK
nasgw9_2_Disk_0  OK        NOT_CONN  NOT_CONN  NOT_CONN  NOT_CONN  NOT_CONN
nasgw9_1_Disk_1  OK        OK        OK        OK        OK        OK
EMC_CLARiiON0_0  OK        OK        OK        OK        OK        OK
EMC_CLARiiON0_1  OK        OK        OK        OK        OK        OK
```

Display the list of disks and their properties.

```
Storage> disk list detail
Disk             Pool             Enclosure   Array Type    Size (Use%)   Transport
↪ID                 Serial Number
====             ==============   ==========  ===========   ============  ==========
↪==================  =============
ams_wms0_10      *coordinator*    ams_wms0    A/A           1.00G 0.0%    FC
↪HITACHI:DF600F:4:1  71011588000A
ams_wms0_11      *coordinator*    ams_wms0    A/A           1.00G 0.0%    FC
↪HITACHI:DF600F:4:2  71011588000B
ams_wms0_12      p03              ams_wms0    A/A           1.00G 27.5%   FC
↪HITACHI:DF600F:4:3  71011588000C
ams_wms0_13      p03              ams_wms0    A/A           1.00G 18.8%   FC
↪HITACHI:DF600F:4:4  71011588000D
ams_wms0_14      p04              ams_wms0    A/A           1.00G 17.8%   FC
↪HITACHI:DF600F:4:5  71011588000E
```

Show the list of multiple paths of disks connected to all the nodes of the cluster. Also show the status of those paths on each node in the cluster.

```
Storage> disk list paths
Disk             Paths    nasgw78_2              nasgw78_1
====             =====    ========               ========
AMS_WMS0_0       Path 1   primary,enabled,active primary,enabled,active
AMS_WMS0_1       Path 1   primary,enabled,active primary,enabled,active
nasgw78_2_Disk_0 Path 1   enabled,active         -
nasgw78_1_Disk_1 Path 1   -                      enabled,active
```

Grow a selected disk.

```
Storage> disk grow Disk_0
ACCESS disk SUCCESS V-288-0 disk grow Disk_0 completed successfully.
```

Format a specified disk.

```
Storage> disk format aluadisk0_7
You may lose all the data on the disk, do you want to continue (y/n, the default is␣
↪n):y
ACCESS disk ERROR V-288-832 Disk aluadisk0_7 is used by pool pool_1, remove the disk␣
↪from the use by
command  pool rmdisk  firstly and then format.
ACCESS disk INFO V-288-832  File System fs_1 is based on the disk aluadisk0_7.
ACCESS disk INFO V-288-832  File System fs_2 is based on the disk aluadisk0_7.
ACCESS disk INFO V-288-832  File System fs_4 is based on the disk aluadisk0_7.
```

Remove a disk from the configuration.

```
Storage> disk remove aluadisk0_7
Removed disk emc0_03ff from nasgw78_1
Removed disk emc0_03ff from nasgw78_2
```

Destroy the coordinator pool.

```
Storage> disk format aluadisk0_7, aluadisk0_8
You may lose all the data on the disk, do you want to continue (y/n, the default is␣
↪n):y

ACCESS Disk SUCCESS V-288-0 disk format: aluadisk0_7 has been formatted successfully.

ACCESS Disk ERROR V-288-0 Disk aluadisk0_8 has invalid pool configuration. Format␣
↪with force option to clean up else run scanbus to retrieve.
Storage> disk format aluadisk0_8 force
You may lose all the data on the disk, do you want to continue (y/n, the default is␣
↪n):y

ACCESS Disk SUCCESS V-288-0 disk format: aluadisk0_8 has been formatted successfully.
Storage> fencing destroy
```

Enable I/O fencing on the disks `ams_wms0_000d`, `ams_wms0_000e`, `ams_wms0_000f` (only three disks required).

```
Storage> fencing on disk ams_wms0_000d,ams_wms0_000e,ams_wms0_000f
ACCESS fencing Success V-288-0 IO Fencing feature now Enabled with SCSI3 Persistent␣
↪Reservations
100% [#] Enabling fencing
```

Check I/O fencing status after enabling fencing.

```
Storage> fencing status

IO Fencing Status
=================
Enabled
Disk Name           Coord Flag On
==============      ==============
ams_wms0_000d       Yes
ams_wms0_000e       Yes
ams_wms0_000f       Yes
```

Disable I/O fencing on all the nodes.

```
Storage> fencing off
ACCESS fencing Success V-288-0 IO Fencing feature now Disabled
100% [#] Disabling fencing
```

Check I/O fencing status after disabling I/O fencing.

```
Storage> fencing status
IO Fencing Status
=================
Disabled
Disk Name           Coord Flag On
==============      ==============
ams_wms0_000d       Yes
ams_wms0_000e       Yes
ams_wms0_000f       Yes
```

Replace ams_wms0_000f with ams_wms0_000c with fencing in Enabled mode.

```
Storage> fencing replace ams_wms0_000f ams_wms0_000c
ACCESS fencing Success V-288-0 Replaced disk ams_wms0_000f with ams_wms0_000c␣
↪successfully.
100% [#] Replacing disk ams_wms0_000f with ams_wms0_000c
```

Replace ams_wms0_000f with ams_wms0_000c with fencing in Disabled mode.

```
Storage> fencing replace ams_wms0_000f ams_wms0_000c
ACCESS fencing Success V-288-0 Replaced disk ams_wms0_000f with ams_wms0_000c␣
↪successfully.
Please run fencing on.
100% [#] Replacing disk ams_wms0_000f with ams_wms0_000c
```

Check I/O fencing status after the I/O fencing replace operation is completed.

```
Storage> fencing status
IO Fencing Status
=================
Disabled
Disk Name           Coord Flag On
==============      ==============
ams_wms0_000c       Yes
ams_wms0_000d       Yes
ams_wms0_000e       Yes
```

Enable I/O fencing on all of the nodes without specifying the disks. This will work only if the user had specified three disks earlier for I/O fencing.

```
Storage> fencing on disk
ACCESS fencing Success V-288-0 IO Fencing feature now Enabled with SCSI3 Persistent␣
↪Reservations
100% [#] Enabling fencing
```

Check I/O fencing status after enabling I/O fencing again.

```
Storage> fencing status
IO Fencing Status
=================
Enabled
Disk Name           Coord Flag On
```

```
==============      ==============
ams_wms0_000c       Yes
ams_wms0_000d       Yes
ams_wms0_000e       Yes
```

Create a mirrored file system with name `fs1` with disks from the pools `pool1` and `pool2`.

```
Storage> fs create mirrored fs1 100M 2 pool1,pool2
100% [#] Creating mirrored filesystem
```

Defragment a file system now.

```
Storage> storage fs defrag now fs_t 2M
Do you want to continue? yes|no
yes
```

You can check defrag status by running fs list command

```
Storage> storage fs list fs_t
General Info:
===============
Block Size:       8192 Bytes
Version:          Version 10
phoenix_01:        online
phoenix_02:        online

Primary Tier
============
Size:            100.00M
Use%:            6%
Layout:          simple
Mirrors:         -
Columns:         -
Stripe Unit:     0.00 K
Meta Data:       metaOk
FastResync:      Disabled

1. Mirror 01:
List of pools:   pool1
List of disks:   vmdk0_4

FS Type:           Normal

Defrag Status: Done successfully
Fullfsck Status: Not Running
Resync   Status: Not Running
Rollsync Status: Not Running
Relayout Status: Not Running
```

Create a file system for a virtual machine workload with 4 striped columns.

```
Storage> fs create pretuned vmdk_fs 100g pool1  workload=virtualmachine␣
↪layout=striped 4
```

Create a file system for a media server workload with 8 striped columns.

```
Storage> fs create pretuned media_fs 100g pool2  workload=mediaserver layout=striped-
↪mirror 8
```

Create an ecoded file system with stripe unit as 64k and with 3 data columns and 1 parity column and failure domain as node. Striped aligned is set to no and rotating parity is set to yes by default for ecoded NFS use case.

```
Storage> fs create ecoded ecfs1 4g 3 1 pool1 blksize=1024 64k stripe_aligned=no␣
↪stripe_tag=node rotating_parity=yes workload=virtualmachine

Following are the ecoded specific parameters:
stripe_unit:       stripe width size to use for ecoded volume
stripe_aligned:    object allocations are aligned to stripe length
stripe_tag:        tag for customize failure domain(disk/node)
rotating_parity:   striping with distributed parity.
workload:          type of workload using this file system
```

List the file system properties of vmdk_fs.

```
Storage> fs list vmdk_fs
General Info:
===============
Block Size:       8192 Bytes
Version:          Version 11
Workload:         virtualmachine datastore
Extent Size:      1m
IAS_01:        online
IAS_02:        online

Primary Tier
============
Size:            100G
Use%:            5%
Layout:          striped
Mirrors:         -
Columns:         4
Stripe Unit:     512 K
Metadata:        metaOk
FastResync:      Disabled
```

Add a mirror to file system fs1.

```
Storage> fs addmirror fs1 pool3,pool4
```

Remove a mirror of file system fs1 that is residing on Disk AMS_WMS0_0.

```
Storage> fs rmmirror fs1 AMS_WMS0_0
```

Add two columns to file system fs1.

```
Storage> fs addcolumn fs1 2 pool3
```

Remove column from file system fs1.

```
Storage> fs rmcolumn fs1
```

Show file systems that have stale mirrors.

```
Storage> fs checkmirror
fs_name:
----------
mirror3
```

Resynchronize all stale mirrors for all file systems.

```
Storage> fs resync
Resync stale mirror for file systems are started in background.
```

Resynchronize all stale mirrors for a particular file system.

```
Storage> fs resync mirror3
Resync stale mirror for file system mirror3 is started in background.
```

Show resynchronization progress running in background.

```
Storage> fs checkresync
FS                  MIRROR              TYPE      PROGRESS  START_TIME              ␣
↪USED_TIME   REMAINING_TIME
=================================================================================================
mir2                tier 1,mirror 02    RESYNC    6.46%     Jun/05/2011/09:39:53    ␣
↪0:5:9       1:14:34
mir3_roll2          tier 1          ROLLBACK  1.28%     Jun/05/2011/14:51:40    ␣
↪0:0:12      15:23
mir3                tier 1,mirror 03    RESYNC    7.67%     Jun/05/2011/15:10:26    ␣
↪0:1:14      14:50
```

Make a file system offline.

```
Storage> fs offline fs1
100% [#] Offline filesystem
```

Check and repair a file system while it is in offline state.

```
Storage> fs fsck fs1
```

Make a file system online.

```
Storage> fs online fs1
100% [#] Online filesystem
```

Check and repair a file system while it is in offline state.

```
Storage> fs fsck fs1
ACCESS fs ERROR V-288-693 fs1 must be offline to perform fsck.
```

Grow file system's primary tier size to 1 G.

```
Storage> fs growto primary fs1 1g
```

Grow file system's primary tier size by 50 M.

```
Storage> fs growby primary fs1 50M
```

Shrink file system's primary tier by the specified size.

```
Storage> fs shrinkby primary fs1 10m
```

Shrink file system's primary tier to just 50 M.

```
Storage> fs shrinkto primary fs1 50M
```

View the list of file systems.

```
Storage> fs list
FS                          STATUS       SIZE    LAYOUT     MIRRORS   COLUMNS   USE%
↪NFS SHARED  CIFS SHARED  SECONDARY TIER
========================== ======       ====    ======     =======   =======   ====
↪==========  ===========  ==============
fs1                         online    100.00M    simple        -         -          3%
↪    no          no           no
fs2                         online    100.00M    simple        -         -          3%
↪    no          no           no
fs3                         online    700.00M    simple        -         -          4%
↪    no          no           no
fs4                         online     69.00M    simple        -         -          4%
↪    no          no           no
fs5                         online      1.94G    simple        -         -          1%
↪    no          no           no
```

Display detailed information for a specific file system.

```
Storage> fs list mir4
General Info:
===============
Block Size:       1024 Bytes
Version:          Version 8
Cluster5_01:        online

Primary Tier
============
Size:             30.00G
Use%:             0%
Layout:           mirrored
Mirrors:          4
Columns:          -
Stripe Unit:      0.00 K
FastResync:       Enabled

1. Mirror 01:
List of pools:   pool1
List of disks:   disk1 disk2

2. Mirror 02:
List of pools:   pool1
List of disks:   disk3 disk4

3. Mirror 03:
List of pools:   pool1
List of disks:   disk5 disk6

4. Mirror 04:
List of pools:   pool1
List of disks:   disk7 disk8

Secondary Tier
==============
Size:             20.00G
Use%:             0%
Layout:           mirrored
Mirrors:          2
```

```
Columns:         -
Stripe Unit:     0.00 K
FastResync:      Disabled

1. Mirror 01:
List of pools:   pool1
List of disks:   disk9 disk10

2. Mirror 02:
List of pools:   pool1
List of disks:   disk11 disk12

FS Type:           Normal

Defrag Status: Not Running
Fullfsck Status: Not Running
Resync Status:
    Tier 1, Mirror 03: 7.67%     Start_time: Jun/05/2011/15:10:26   Work_time: 0:1:14␣
↪    Remaining_time: 14:50
    Tier 1, Mirror 04: 1.08%     Start_time: Jun/05/2011/13:08:07   Work_time: 0:1:21␣
↪    Remaining_time: 2:03:33
    Tier 2, Mirror 02: 11.27%    Start_time: Jun/05/2011/14:03:14   Work_time: 0:0:18␣
↪    Remaining_time: 02:21
Rollsync Status:
    Rollsync  mir4_roll, Tier 1: 1.28%  Start_time: Jun/05/2011/14:51:40   Work_time:␣
↪0:0:12     Remaining_time: 15:23
Relayout Status: Not Running
```

Disable FastResync for a file system.

```
Storage> fs unsetfastresync fs6
```

Try disabling FastResync for a file system where it is already in the disabled state.

```
Storage> fs unsetfastresync fs6
ACCESS fs ERROR V-288-655 Fastresync is not enabled for fs6.
```

Enable FastResync for a file system.

```
Storage> fs setfastresync fs6
```

Try enabling FastResync for a file system where it is already in the enabled state.

```
Storage> fs setfastresync fs6
ACCESS fs ERROR V-288-651 File system fs6 is already fastresync enabled.
```

Destroy a file system.

```
Storage> fs destroy fs6
100% [#] Destroy filesystem
FS ============
fs1
```

Display WWN information for a particular node

```
Storage> hba democluster_01
Node            Host Initiator HBA WWNs
```

```
====            ========================
democluster_01      21:00:00:1b:32:1e:5c:ba, 21:01:00:1b:32:3e:5c:ba
```

Display WWN information for all the running nodes in the cluster

```
Storage> hba
Node            Host Initiator HBA WWNs
====            ========================
democluster_01      21:00:00:1b:32:1e:5c:ba, 21:01:00:1b:32:3e:5c:ba
democluster_02      21:00:00:e0:8b:92:53:d7, 21:01:00:e0:8b:b2:53:d7
```

Create a pool `pool1` with disks `Disk_0`, `Disk_1`.

```
Storage> pool create pool1 Disk_0,Disk_1 isolated=yes
ACCESS pool Success V-288-1015 Pool pool1 created successfully.
100% [#] Creating pool pool1
```

View the list of pools.

```
Storage> pool list
Pool  Pool Type  List of disks
=====  ========  ==============
pool1  Isolated  Disk_0 Disk_1
pool2  Normal    Disk_2 Disk_5
```

Destroy pool `pool1`.

```
Storage> pool destroy pool1
ACCESS pool SUCCESS V-288-2056 successfully destroyed the pool.
```

Remove disks from a pool.

```
Storage> pool rmdisk Disk_2
ACCESS pool Success V-288-1360 Disk(s) Disk_2 have been removed successfully.
```

Add disk `Disk_2` to pool `pool2`.

```
Storage> pool adddisk pool2 Disk_2
ACCESS pool Success V-288-1001 Disk(s) Disk_2 are added to pool2 successfully.
```

Rename `pool1` to `p01`.

```
Storage> pool rename pool1 p01
ACCESS pool Success V-288-1017 Pool rename successful.
```

Move disk `Disk_0` from pool `p01` to pool `pool2`.

```
Storage> pool mvdisk p01 pool2 Disk_0
ACCESS pool Success V-288-1002 Disk(s) moved successfully.
```

View the free space in each of the pools.

```
Storage> pool free
Pool                                    Free Space     Total Space    Use%
==================================      ==========     ===========    ====
p01                                     989.64M         989.64M        0%
pool2                                   2.90G           2.90G          0%
```

View the free space in `pool2`.

```
Storage> pool free pool2
Disk              Free Space     Total Space    Use%
====              ==========     ===========    ====
Disk_0            989.64M        989.64M          0%
Disk_2            991.69M        991.69M          0%
Disk_5            991.69M        991.69M          0%
```

Mark disk Disk_2 in `pool2` as spare disk.

```
Storage> pool markdiskspare pool_name disk1[,disk2,...]
Storage> pool markdiskspare pool2 Disk_2
ACCESS Pool SUCCESS V-288-1360 Disk(s) Disk_2 have been marked as spared disks␣
↪successfully.
```

Remove spare disk flag on Disk_2 in `pool2`.

```
Storage> pool removediskspare pool_name disk1[,disk2,...]
Storage> pool removediskspare pool2 Disk_2
ACCESS Pool SUCCESS V-288-0 Disk(s) Disk_2 have been removed as spared disks␣
↪successfully.
```

Scan SCSI bus for newly added disks.

```
Storage> scanbus
100% [#] Scanning the bus for disks
```

Create a snapshot of a file system.

```
Storage> snapshot create snapshot1 fs1
100% [#] Create snapshot
```

Restore a file system by a given snapshot.

```
Storage> snapshot restore fs0 snap00
```

Destroy a snapshot of an existing file system.

```
Storage> snapshot destroy snap1 fs1
100% [#] Destroy snapshot
```

View the list of snapshots.

```
Storage> snapshot list
Snapshot                                  FS       Status     ctime                 ␣
↪mtime                  Removable  Preserved
=======                                   ===      ======     =====                 ␣
↪=====                  =========  =========
snap2                                     fs1      offline    2009.Jul.27.02:40:43  ␣
↪2009.Jul.27.02:40:57   no         no
sc1_24_Jul_2009_21_34_01_IST              fs1      offline    2009.Jul.24.21:34:03  ␣
↪2009.Jul.24.21:34:03   yes        no
sc1_24_Jul_2009_19_34_02_IST              fs1      offline    2009.Jul.24.19:34:04  ␣
↪2009.Jul.24.19:34:04   yes        no
presnap_sc1_24_Jul_2009_18_34_02_IST      fs1      offline    2009.Jul.24.18:34:04  ␣
↪2009.Jul.24.18:34:04   yes        yes
sc1_24_Jul_2009_17_34_02_IST              fs1      offline    2009.Jul.24.17:34:04  ␣
↪2009.Jul.24.17:34:04   yes        no
```

Display the list of snapshots for a given file system.

```
Storage> snapshot list fs2
Snapshot                              Schedule Name   Status    ctime                ␣
↪mtime                Removable   Preserved
======                                =============   ======    =====                ␣
↪=====                =========   =========
snap2                                       -          offline   2009.Jul.27.02:40:43 ␣
↪2009.Jul.27.02:40:57  yes        no
sc1_24_Jul_2009_22_34_02_IST                sc1        offline   2009.Jul.24.22:34:09 ␣
↪2009.Jul.24.22:34:09  yes        no
sc1_24_Jul_2009_21_34_01_IST                sc1        offline   2009.Jul.24.21:34:03 ␣
↪2009.Jul.24.21:34:03  yes        no
sc1_24_Jul_2009_19_34_02_IST                sc1        offline   2009.Jul.24.19:34:04 ␣
↪2009.Jul.24.19:34:04  yes        no
presnap_sc1_24_Jul_2009_18_34_02_IST        -          offline   2009.Jul.24.18:34:04 ␣
↪2009.Jul.24.18:34:04  yes        yes
sc1_24_Jul_2009_17_34_02_IST                sc1        offline   2009.Jul.24.17:34:04 ␣
↪2009.Jul.24.17:34:04  yes        no
```

Display the list of snapshots corresponding to a file system and schedule name.

```
Storage> snapshot list fs1 sc1
Snapshot                                    Status     ctime                     mtime  ␣
↪                   Removable
================================            ======     =====                     =====  ␣
↪                   =========
sc1_24_Jul_2009_22_34_02_IST                offline    2009.Jul.24.22:34:09      2009.
↪Jul.24.22:34:09      yes
sc1_24_Jul_2009_21_34_01_IST                offline    2009.Jul.24.21:34:03      2009.
↪Jul.24.21:34:03      yes
sc1_24_Jul_2009_20_34_02_IST                offline    2009.Jul.24.20:34:04      2009.
↪Jul.24.20:34:04      yes
sc1_24_Jul_2009_19_34_02_IST                offline    2009.Jul.24.19:34:04      2009.
↪Jul.24.19:34:04      yes
sc1_24_Jul_2009_18_34_02_IST                offline    2009.Jul.24.18:34:04      2009.
↪Jul.24.18:34:04      yes
```

Make a snapshot offline.

```
Storage> snapshot offline snapshot1 fs1
100% [#] Offline snapshot
```

Make a snapshot online.

```
Storage> snapshot online snapshot1 fs1
100% [#] Online snapshot
```

Enable the snapshot quota of a file system.

```
Storage> snapshot quota on fs1
Storage> snapshot quota on fs1 1M
```

Disable the snapshot quota of a file system.

```
Storage> snapshot quota off fs2
Storage> snapshot quota off fs3
```

```
Storage> snapshot quota off fs4
Storage> snapshot quota off fs5
```

Display the list of snapshot quotas of all the file systems.

```
Storage> snapshot quota list
FS                                  Quota      Capacity Limit
=================================== =====      ==============
fs1                                 on              1M
fs2                                 off              0
fs3                                 off              0
fs4                                 off              0
fs5                                 off              0
```

Create a schedule for automated snapshot creation of a given file system every 3 hours on a daily basis without any restrictions on the maximum number of snapshots.

```
Storage> snapshot schedule create schedule1 fs1 0 * 3 * * *
```

Modify the existing schedule so that the snapshot is created every 2 hours on the first day of the week and only 20 snapshots can be created for a given schedule.

```
Storage> snapshot schedule modify schedule1 fs1 20 * 2 * * 1
```

Destroy all automated snapshots created under a given schedule and file system.

```
Storage> snapshot schedule destroyall schedule1 fs1
```

Preserve a snapshot created according to a given schedule and file system name.

```
Storage> snapshot schedule preserve schedule1 fs1 schedule1_Feb_27_16_42_00_IST
```

List all schedules created for automated snapshot creation corresponding to an existing file system.

```
Storage> snapshot schedule show fs3
FS            Schedule Name              Max Snapshot     Minute     Hour    Day
↪Month   WeekDay
===========   ========================  =============     ======     ====    ===
↪=====   =======
fs3           sched1                          0            */2        *       *
↪*        *
fs3           sched2                          20           */45       *       *
↪*        *
```

List automated snapshot schedules for all the file systems.

```
Storage> snapshot schedule show
FS           Schedule Name               Max Snapshot     Minute     Hour    Day
↪Month   WeekDay
===========  =========================  =============     ======     ====    ===
↪=====   =======
.b
fs6          sc1                             10           */5        *       *
↪*        *
fs1          sc1                             10           */25       *       *
↪*        *
```

Delete all schedules created for automated snapshot creation / removal corresponding to an existing file system.

```
Storage> snapshot schedule delete fs1
```

Create a mirrored file system with name `fs1` with disks from the pools `pool1` and `pool2`. Add a mirrored tier to this file system.

```
Storage> fs create mirrored fs1 100M 2 pool1,pool2
100% [#] Creating mirrored filesystem
Storage> tier add mirrored fs1 100M 2 pool3,pool4
100% [#] Creating mirrored secondary tier of filesystem
```

Create a simple file system on encrypted volume

```
Storage> fs create simple fs4 1g pool1 blksize=2048 pdir_enable=no encrypt=on
100% [#] Creating simple filesystem
ACCESS fs SUCCESS V-288-0 Created simple file system fs4
```

Display detailed information for a specific file system

```
Storage> fs list fs4
General Info:
===============
Block Size:        2048 Bytes
Version:           Version 11
Volume Encrypted:  Yes
ISA_01:        online
ISA_02:        online

Primary Tier
============
Size:            1.00G
Use%:            5%
Layout:          simple
Mirrors:         -
Columns:         -
Stripe Unit:     0.00 K
Meta Data:       metaOk
FastResync:      Disabled

1. Mirror 01:
List of pools:   pool1
List of disks:   isa_01_intel_nvme2_0

FS Type:           Normal

Defrag Status: Not Running
Fullfsck Status: Not Running
Resync   Status: Not Running
Rollsync Status: Not Running
Relayout Status: Not Running

WORM Enabled: No
```

Create a simple WORM enabled file system

```
Storage> fs create simple fs1 1g pool1 blksize=2048 pdir_enable=no encrypt=off␣
↪worm=yes
100% [#] Creating simple filesystem
```

```
ACCESS fs SUCCESS V-288-0 Created simple file system fs1
```

Display detailed information for a specific file system

```
Storage> fs list fs1
General Info:
===============
Block Size:        2048 Bytes
Version:           Version 13
Volume Encrypted:  No
Max IOPS:  0
ISA_01:        online
ISA_02:        online

Primary Tier
============
Size:            1.00G
Use%:            5%
Layout:          simple
Mirrors:         -
Columns:         -
Stripe Unit:     0.00 K
Meta Data:       metaOk
FastResync:      Disabled

1. Mirror 01:
List of pools:   pool1
List of disks:   isa_01_intel_nvme2_0

FS Type:           Normal

Defrag Status: Not Running
Fullfsck Status: Not Running
Resync   Status: Not Running
Rollsync Status: Not Running
Relayout Status: Not Running

WORM Enabled: Yes
```

Add a new AWS service with name *amzservice*:

```
Storage> cloud addservice amzservice service_provider=AWS
Access_Key: ZTVhY2E3M2JlYWQyNWFk
Secret_Key:
```

Add an S3-compatible cloud service provider with name *isaobj*:

```
Storage> cloud addservice isaobj service_provider=S3_COMPATIBLE
REST Endpoint: s3.isaobj:8143
Access_Key: ZTVhY2E3M2JlYWQyNWFk
Secret_Key:
Successfully added new cloud service.
```

Add the cloud storage of the Amazon S3 service, named awstest, in the region of us-west-2 as the cloud tier cloudtier for the scale-out file system largefs1.

```
Storage> tier add cloud largefs1 cloudtier awstest us-west-2
ACCESS tier SUCCESS V-288-0 Tier cloudtier is added for largefs1
```

Add S3-compatible cloud storage, named s3comptest, in the region of none as the cloud tier for the scale-out file system largefs1. Use none as the region for S3-compatible cloud storage.

```
Storage> tier add cloud largefs1 va01 s3comptest none
ACCESS tier SUCCESS V-288-0 Tier va01 is added for largefs1
```

List all the configured cloud services.

```
Storage> cloud listservice
Service     Provider        Access Key     REST Endpoint
==========  =========       ============   ==============
amzservice  AWS             AKIA----B4WA   -
isaobj      S3_COMPATIBLE   ZTVh----NWFk   s3.isaA:8143
```

Remove an existing cloud service named *amzservice*:

```
Storage> cloud removeservice amzservice
```

Remove an existing cloud service named *isaobj*:

```
Storage> cloud removeservice isaobj
```

Add a mirror to the secondary tier of the file system `fs1`.

```
Storage> tier addmirror fs1 pool5
100% [#] Adding mirror to secondary tier of filesystem
```

Add a column to the tier of the file system.

```
Storage>  tier addcolumn fs1 1 pool1
ACCESS fs INFO V-288-2644 This operation may take some time to complete. You can␣
↪check the status
in storage fs list fsname
```

Remove a column from the tier of the file system.

```
Storage> tier rmcolumn fs1
ACCESS fs INFO V-288-2711 Only one column will be removed.
ACCESS fs INFO V-288-2644 This operation may take some time to complete. You can␣
↪check the status
in storage fs list fsname
```

Show list of files on the primary tier of file system fs1.

```
Storage> tier listfiles fs1 primary
/.placement_policy.xml
/lost+found/changelog
```

Show list of files on the secondary tier of file system fs1.

```
Storage> tier listfiles fs1 secondary
```

Show tier location of file a.txt, which is in the root directory of file system fs1.

```
Storage> tier mapfile fs1 /a.txt
Tier        Extent Type   File Offset   Extent Size
====        ===========   ===========   ===========
Primary     Data          0 Bytes       1.00 KB
```

The following command sets the policy of file system fs1, such that the new files get created on the primary tier. If a file has not been accessed for more than 7 days, then it is moved from the primary to the secondary tier. If the access temperature of the files in the secondary tier is more than 5, then these files are moved from the secondary tier to the primary tier. Access temperature is calculated over a 3-day period. The prune policy is to show the file system can delete the inactive files on the secondary tier after the specified time.

Modify the tier policy.

```
Storage> tier policy modify fs1 primary 7 5 3
ACCESS fs SUCCESS V-288-0 Successfully modified tiering policy for File system fs1
```

List the tiering policy for all the file systems.

```
Storage> tier policy list
FS                        Create on   Days  MinAccess Temp  PERIOD
========================= =========   ====  ==============  ======
fs1                       primary       7          5          3
```

Modify the prune policy.

```
Storage> tier policy prune modify fs1 180
ACCESS fs SUCCESS V-288-0 Successfully modified the Prune policy for File system fs1
```

List the tiering prune policy for all the file systems.

```
Storage> tier policy prune modify fs1 180
FS                        Delete After
========================= =========
fs1                       180
```

Run the tiering policy for file system fs1.

```
Storage> tier policy run fs1
ACCESS fs SUCCESS V-288-0 Successfully ran tiering policy for File system fs1
```

Show list of files that will be moved and/or deleted by running the policy.

```
Storage> tier query fs1
/a.txt
/b.txt
/c.txt
/d.txt
```

Set schedule of file system fs1.

```
Storage> tier schedule modify fs1 1 1 1 * *
ACCESS fs SUCCESS V-288-0 Command 'tier schedule modify' executed sucessfully for fs1
```

Get the tiering schedule for all the file systems.

```
Storage> tier schedule list
FS                          Minute     Hour      Day    Month  WeekDay  NodeName
==========================  ======     ====      ===    =====  =======  ========
fs1                              1        1        1        *        *  master node
```

Relocate a file from the secondary tier to the primary tier of file system fs1.

```
Storage> tier schedule relocate fs1 /a.txt
ACCESS fs SUCCESS V-288-0 Successfully relocated /a.txt from secondary to primary␣
↪tier for File system fs1
```

Remove the tiering schedule for file system fs1.

```
Storage> tier schedule remove fs1
ACCESS fs SUCCESS V-288-0 Command tier schedule remove executed sucessfully for fs1
```

Remove the policy of file system fs1.

```
Storage> tier policy remove fs1
ACCESS fs SUCCESS V-288-0 Successfully removed tiering policy for File system fs1
```

Remove the prune policy of file system fs1.

```
Storage> tier policy prune remove fs1
ACCESS fs SUCCESS V-288-0 Successfully removed the Prune policy for File system fs1
```

Remove a mirror from the secondary tier of file system fs1.

```
Storage> tier rmmirror fs1
```

Remove the cloud tier cloudtier from scale-out file system largefs1.

```
Storage> tier remove largefs1 cloudtier
ACCESS tier SUCCESS V-288-0 Tier cloudtier is removed
```

Set fastresync for the tier of the file system.

```
Storage> tier setfastresync adistr pool1
Storage>
```

Unset fastresync for the tier of the file system.

```
Storage> tier unsetfastresync adistr
ACCESS fs SUCCESS V-288-1817 Disabled fast resync on the file system adistr
```

Show the access statistics of the cloud tier cloudtier.

```
Storage> tier stats show largefs1 cloudtier
GET       168
GET bytes 174.5MB
PUT       918
PUT bytes 10.3GB
DELETE    20
```

Monitor the access statistics of the cloud tier cloudtier every 10 seconds.

```
Storage> tier stats monitor largefs1 cloudtier 10
        GET     GET bytes       PUT     PUT bytes       DELETE
        0       0               0       0               0
        1       1.4MB           0       0               0
        0       0               3       714.0KB         0
        0       0               0       0               0
```

Reset the access statistics of the cloud tier cloudtier to zero. After executing the tier stats reset command, the output for the tier stats show command is reset to zero.

```
Storage> tier stats reset largefs1 cloudtier
```

Show the usage of the cloud tier cloudtier.

```
Storage> tier stats usage largefs1 cloudtier
Storage Utilized     223.1GB
Number of objects    488
Number of files      231
```

Enable deduplication on the file system *fs1*.

```
Storage> dedup enable fs1 blksize=8192
```

Set the deduplication schedule.

```
Storage> dedup schedule set fs1 0,6,12,18 2 3
```

List deduplication enabled file systems.

```
Storage> dedup list
Default  Priority  CPU    Memory
--------------------------------
         NORMAL    IDLE     128M
Filesystem                Priority  CPU    Granularity  Enabled  Schedule
↪NodeList
-----------------------------------------------------------------------------------
↪---------
/vx/fs1                   NORMAL    IDLE         8192B  YES      SET
↪node_01,node_02
```

Show deduplication parameters details of file system fIfs1fp.

```
Storage> dedup list fs1
Priority               NORMAL
CPU                    IDLE
Memory                 128M
Granularity            8192B
Enabled                YES
Schedule hours         Every hour
Schedule day interval  Every Sunday
Schedule frequency     1
NodeList               node_01,node_02
```

Check whether dedup dry run reaches to threshold value 60%.

```
Storage> dedup dryrun fs1 60
```

Show deduplication status.

```
Storage> dedup status
Filesystem                Saving  Status     Node             Type       Details
-----------------------------------------------------------------------------------
↪--------
 /vx/fs1                     00%  COMPLETED  node_01          MANUAL     2011/05/23
↪17:08:00 Dryrun
threshold not met. Cleaning up...
```

Start deduplication on file system *fs1*.

```
Storage> dedup start fs1
```

Start deduplication on file system *fs1* at node *node_01*.

```
Storage> dedup start fs1 node1
```

Show deduplication status.

```
Storage> dedup status
Filesystem                Saving  Status     Node             Type       Details
-----------------------------------------------------------------------------------
↪--------
 /vx/fs1                      --  RUNNING    node_01          MANUAL     --
```

Show status output after completion of deduplication.

```
Storage> dedup status
Filesystem                Saving  Status     Node             Type       Details
-----------------------------------------------------------------------------------
↪--------
 /vx/fs1                     34%  COMPLETED  node_01          MANUAL    2011/05/23␣
↪17:20:53
End detecting duplicates and filesystem changes 0
```

Remove deduplication on file system *fs1*.

```
Storage> dedup remove fs1
```

List deduplication enabled file system.

```
Storage> dedup list
Default  Priority  CPU    Memory
--------------------------------
         NORMAL    IDLE     128M
Filesystem                Priority  CPU    Granularity  Enabled  Schedule          ␣
↪NodeList
-----------------------------------------------------------------------------------
↪---------
```

Create a data movement policy policy1 for file system fs1 to move the files with file name extensions of .txt and .pdf that were not accessed or modified within the last 2 days from the primary (disk tier) to tier1 (cloud tier).

```
Storage> fs policy add operation=move policy1 fs1 primary tier1 retrieval_
↪option=Standard \*.txt,\*.pdf atime >2d mtime >2d
ACCESS policy SUCCESS V-288-0 Policy policy1 for fs fs1 added successfully.
```

Create a data deletion policy policy2 for file system fs1 to delete the files with file name extensions of .txt and .pdf that were not accessed or modified within the last 2 days from tier1 (cloud tier).

```
Storage> fs policy add operation=delete policy2 fs1 tier1 \*.txt,\*.pdf atime >2d␣
↪mtime >2d
ACCESS policy SUCCESS V-288-0 Policy policy2 for fs fs1 added successfully.
```

Modify data movement policy policy1 for file system fs1 to move the files with file name extension of .doc that were not accessed or modified within the last 3 days.

```
Storage> fs policy modify policy1 \*.doc atime >3d mtime >3d
ACCESS policy SUCCESS V-288-0 Policy policy1 modified successfully.
```

List all policies.

```
Storage> fs policy list
Name     FS name   Action   Source Tier      Destination Tier   Retrieval Option  ␣
↪Pattern       Atime   Mtime   State
=======  ========  =======  ===============  =================  =================  ␣
↪============  ======  ======  ===========
policy2  fs1       delete   tier1            -                  Standard           \*.
↪txt, \*.pdf  >2d     >2d     not running
policy1  fs1       move     primary          tier1              Standard           \*.
↪doc         >3d     >3d     not running
```

List all policies set for file system fs1.

```
Storage> fs policy list fs1
Name     FS name   Action   Source Tier      Destination Tier   Retrieval Option  ␣
↪Pattern       Atime   Mtime   State
=======  ========  =======  ===============  =================  =================  ␣
↪============  ======  ======  ===========
policy2  fs1       delete   tier1            -                  Standard           \*.
↪txt, \*.pdf  >2d     >2d     not running
policy1  fs1       move     primary          tier1              Standard           \*.
↪doc         >3d     >3d     not running
```

Delete policy policy1 set for file system fs1.

```
Storage> fs policy delete policy1 fs1
ACCESS policy SUCCESS V-288-0 Policy policy1 for fs fs1 deleted successfully.
```

Rename policy2 to policy3.

```
Storage> fs policy rename policy2 policy3
ACCESS policy SUCCESS V-288-0 Policy policy2 renamed to policy3.
```

Show the status of policy run for the policy *Policy1*.

```
Storage> fs policy status Policy1
Policy Name:                    Policy1
==================================================
Policy Run Type:                normal
Policy Run Status:              running
Total Data (Files):             93.1 GB (100000)
Moved/Deleted Data (Files):     47.7 MB (879)
Last File Visited:              file100.txt
```

Abort the currently running policy *Policy1*.

```
Storage> fs policy abort Policy1
ACCESS policy INFO V-288-0 Policy Policy1 aborted successfully.
```

Start a dry run of the policy *Policy1*.

```
Storage> fs policy dryrun Policy1
ACCESS policy INFO V-288-0 Policy Policy1 dryrun started in background, please
check 'fs policy status' for progress.
```

Pause the currently running policy *Policy1*.

```
Storage> fs policy pause Policy1
ACCESS policy INFO V-288-0 Policy Policy1 paused successfully.
```

Run the currently paused policy *Policy1*.

```
Storage> fs policy run Policy1
Policy Policy1 is not running currently, as it was killed/paused. Would you like to␣
↪start new run (y/n): y
ACCESS policy INFO V-288-0 Policy Policy1 run started in background, please check 'fs␣
↪policy status' for progress.
```

Resume the currently paused policy *Policy1*.

```
Storage> fs policy resume Policy1
ACCESS policy INFO V-288-0 Policy Policy1 resume started in background, please check
↪'fs policy status' for progress.
```

Create the schedule for the file system *lfs1*. The schedule runs every 10 minutes.

```
Storage> fs policy schedule create lfs1 */10 * * * *
ACCESS policy_schedule SUCCESS V-288-0 Schedule create for file system lfs1 done␣
↪successfully.
```

Modify the schedule of the file system *lfs1*. The schedule runs every 20 minutes.

```
Storage> fs policy schedule modify lfs1 */20 * * * *
ACCESS policy_schedule SUCCESS V-288-0 Schedule modify for file system lfs1 done␣
↪successfully.
```

List the schedule of the file system *lfs1*.

```
Storage> fs policy schedule list lfs1
File System Name   Minute   Hour   Day of Month   Month   Day of Week
=================  =======  =====  =============  ======  ============
lfs1               20       *      *              *       *
```

Remove the schedule of the file system *lfs1*.

```
Storage> fs policy schedule remove lfs1
ACCESS policy_schedule SUCCESS V-288-0 Schedule for file system lfs1 removed␣
↪successfully.
```

Set the retention on the file */vx/myfs/file1 5y*.

```
Storage> storage fs retention set /vx/myfs/file1 5y
ACCESS Retention SUCCESS V-288-0 Successfully set retention on /vx/myfs/file1
```

Set the retention on the file */vx/myfs/file1 05-20-2020*.

```
Storage> storage fs retention set /vx/myfs/file1 05-20-2020
ACCESS Retention SUCCESS V-288-0 Successfully set retention on /vx/myfs/file1
```

Set the retention on the directory */vx/myfs/dir1 2y*.

```
Storage> storage fs retention set /vx/myfs/dir1 2y
ACCESS Retention SUCCESS V-288-0 Applying retention on all current files in directory,
↪ this will take some time. Check report showevent for status.
```

Set the retention on the directory */vx/myfs/dir1 05-20-2020*.

```
Storage> storage fs retention set /vx/myfs/dir1 05-20-2020
ACCESS Retention SUCCESS V-288-0 Applying retention on all current files in directory,
↪ this will take some time. Check report showevent for status.
```

Show the retention on the file */vx/myfs/file1*

```
Storage> storage fs retention show /vx/myfs/file1
ACCESS Retention SUCCESS V-288-0 Retention value on file /vx/myfs/file1 is 06-08-2022␣
↪23:12:51
```

Show the retention on the directory */vx/myfs/dir1*

```
Storage> storage fs retention show /vx/myfs/dir1
ACCESS Retention ERROR V-288-0 Specified path is directory path, please give file path
```

Clear the retention on the file */vx/myfs/file1*

```
Storage> storage fs retention clear /vx/myfs/file1
ACCESS Retention SUCCESS V-288-0 Successfully cleared retention on /vx/myfs/file1
```

Clear the retention on the directory */vx/myfs/dir1*

```
Storage> storage fs retention clear /vx/myfs/dir1
ACCESS Retention SUCCESS V-288-0 Clearing retention of all current file in directory,␣
↪this will take some time. Check report showevent for status.
```

Enable worm support for specified file system.

```
Storage> fs worm set fs1
ACCESS fs SUCCESS V-288-0 Enabled WORM for fs1 file system.
```

Disable worm support for specified file system.

```
Storage> fs worm clear fs1
ACCESS fs SUCCESS V-288-0 Disabled WORM for fs1 file system.
```

17.1.5 SEE ALSO

disk(1), hba(1), fencing(1), fs(1), pool(1), scanbus(1), snapshot(1), tier(1), iscsi(1), quota(1) compress(1) maxiops(1)

17.2 cloud

17.2.1 SYNOPSIS

`cloud addservice` *service_name* [service_provider=<service_provider>]

`cloud listservice` [*service_name*]

`cloud removeservice` *service_name*

17.2.2 DESCRIPTION

The storage `cloud` commands manage cloud storage configuration operations. The commands can be used to add, list, and remove user cloud storage subscriptions to Veritas Access.

17.2.3 OPTIONS

service_provider=<service_provider> Specifies the cloud storage provider type. You can use Amazon Web Services (AWS) S3 or Glacier, or any S3-compatible cloud storage as the cloud service provider. If you are using an S3-compatible cloud provider, you need to specify the access key, secret key, and the REST endpoint to the fIservice_providerfp. The REST endpoint is the URL where your S3 server accepts the incoming requests. If an S3-compatible service cannot support AWS signature verion 4, then it can not be added as a cloud service.

`cloud addservice` ***service_name* [*service_provider=<service_provider>*]** Adds a public cloud subscription to Veritas Access as a cloud service with a name. The added subscription is accessed using *service_name* thereafter. You are prompted to provide the subscription credentials. To successfully send requests using AWS, you need to have a valid set of security credentials called access keys. You need to provide the access key and secret key as inputs. Credentials are verified and the service is added.

`cloud listservice` **[*service_name*]** List added cloud services. If *service_name* is provided, then only that service is listed. Otherwise all configured services are listed.

`cloud removeservice` ***service_name*** Remove the service *service_name*.

Warning: Service removal fails if there are active cloud tiers associated with this service at the time of removal.

17.2.4 EXAMPLES

Add a new AWS service with name *amzservice*:

```
Storage> cloud addservice amzservice service_provider=AWS
Access_Key: ZTVhY2E3M2JlYWQyNWFk
Secret_Key:
```

Add an S3-compatible cloud service provider with name *isaobj*:

```
Storage> cloud addservice isaobj service_provider=S3_COMPATIBLE
REST Endpoint: s3.isaobj:8143
Access_Key: ZTVhY2E3M2JlYWQyNWFk
Secret_Key:
Successfully added new cloud service.
```

List all the configured cloud services.

```
Storage> cloud listservice
Service     Provider        Access Key      REST Endpoint
==========  =========       ============    ==============
amzservice  AWS             AKIA----B4WA    -
isaobj      S3_COMPATIBLE   ZTVh----NWFk    s3.isaA:8143
```

Remove an existing cloud service named *amzservice*:

```
Storage> cloud removeservice amzservice
```

Remove an existing cloud service named *isaobj*:

```
Storage> cloud removeservice isaobj
```

17.2.5 SEE ALSO

tier(1), fs(1), disk(1)

17.3 compress

17.3.1 SYNOPSIS

`compress file` *fs_name {file|dir} res_level [alg_strength] [file_pattern]*

`compress list` *fs_name {file|dir}*

`compress modage create` *fs_name* fI *age_rule*

`compress modage remove` *fs_name*

`compress pattern create` *fs_name pattern_rule*

`compress pattern remove` *fs_name*

`compress schedule create` *sched_name duration minute [hour] [day] [month] [day_of_week] [node]*

`compress schedule list` *fs_name*

`compress schedule remove` *sched_name*

`compress schedule show` *[sched_name]*

`compress schedule start` *fs_name sched_name res_level [alg_strength]*

`compress schedule stop` *fs_name*

`uncompress file` *fs_name* fI{file|dir} *res_level*

17.3.2 DESCRIPTION

The `compress` command performs administration functions for the Veritas Access compression feature. The compressed files keep the original name and do not change size, as reported by the `stat` or `ls -l`. However, the number of blocks allocated to the file as reported by the `stat` command or the `ls -s` command may be significantly reduced, depending on the compressibility of the data. The applications can access the files when compress is running, but it is recommended that compression jobs are scheduled during non-peak hours. The compression can be done using [`low` | `medium` | `high`] resource levels. The higher the resource level, the faster the job completes. Compression supports *alg_strength* 1 to 9 values. The default *alg_strength* is 6. Higher *alg_strength* increases storage savings, but it takes more time to complete.

17.3.3 OPTIONS

`compress file` ***fs_name*** **fI{file|dir}** ***res_level [alg_strength] [file_pattern]*** Compress the file or directory of the *fs_name* file system using the specified *res_level* and *alg_strength*. The default *alg-strength* is 6. If the directory is specified, then all the files under the named directory are compressed. For directories, you can specify file name patterns. The *res_level* can be `low`, `medium`, or `high`.

`compress list` ***fs_name {file|dir}*** Lists the compression details for the *fs_name* file system files or directory.

`compress modage create` ***fs_name*** **fI** ***age_rule*** Adds an age-based rule for scheduled compression jobs for the *fs_name* file system. The *age_rule* is specified in days.

`compress modage remove` ***fs_name*** Removes the age-based rule for the *fs_name* file system.

`compress pattern create` ***fs_name pattern_rule*** Adds a pattern-based rule for scheduled compression jobs for the *fs_name* file system.

`compress pattern remove` ***fs_name*** Removes the pattern-based rule for the *fs_name* file system.

compress schedule create ***sched_name duration minute [hour] [day] [month] [day_of_week] [node]*** Create a compression *sched_name* schedule. The time and frequecy of this schedule are specified using the command line in crontab format. This scheduled job runs on the specified *[node]*. If this *node* is not online at the time of the job, then this particular compression job runs on the CFS primary.

compress schedule list ***fs_name*** List the scheduled compression job details for the *fs_name* file system.

compress schedule remove ***sched_name*** Remove the compression *sched_name* schedule. Make sure none of the file systems have this *sched_name* assigned.

compress schedule show ***[sched_name]*** Show the compression *sched_name* schedule information. If *sched_name* is not specified, all the information about all the schedules are displayed.

compress schedule start ***fs_name sched_name res_level [alg_strength]*** Assign or start the compression *sched_name* schedule for the *fs_name* file system. The compression job uses resource and algorithm strength specified in arguments.

compress schedule stop ***fs_name*** Stop the scheduled compression schedule for the *fs_name* file system.

uncompress file ***fs_name*** **fI{file|dir}** ***res_level*** Uncompress the *fs_name* file system file or directory using the specified *res_level*.

17.3.4 EXAMPLES

Compress file data1.dbf in file system tpcc_arch1 using algrorithm strength of `8` and `high` resource level.

```
Storage> compress file tpcc_arch1 data1.dbf high 8
```

Compress all files with pattern *.arch in directory test_dir on the tpcc_arch1 file system using the `low` resource level and algorithm strength of `3` .

```
Storage>compress file tpcc_arch1 test_dir low 3 \*.arch
```

List and display compression information for the files under the test_dir directory.

```
Storage>compress list tpcc_arch1 test_dir
%Comp   Physical    Logical  %Exts  Alg-Str  BSize  Filename
  92%   15.63 MB   200.0 MB   100%   gzip-3  1024k  test_dir/tpcc3.arch
  87%   25.79 MB   200.0 MB   100%   gzip-3  1024k  test_dir/tpcc2.arch
  89%    21.5 MB   200.0 MB   100%   gzip-3  1024k  test_dir/tpcc1.arch
   0%   200.0 MB   200.0 MB     0%        -      -  test_dir/redo3.log
   0%   200.0 MB   200.0 MB     0%        -      -  test_dir/redo2.log
   0%   200.0 MB   200.0 MB     0%        -      -  test_dir/redo1.log
```

Uncompress all the files in test_dir directory with the `medium` resource level.

```
Storage> uncompress file tpcc_arch1 test_dir medium
```

Create a comp_sched_1 schedule that runs the compression job at 1:00 am every Saturday. This compression job should run only for 3 hours.

```
Storage> compress schedule create comp_sched_1 3 0 1 * * 6
```

Show the compression schedule details of the comp_sched_1 schedule.

```
Storage> compress schedule show comp_sched_1
Name            Node       Duration  Minute Hour   Day      Month    WeekDay
====            ====       ========  ====== ====   ===      =====    ========
comp_sched_1    any        3 hours   0      1      *        *        6
```

Start compression schedule for the tpcc_arch1 file system with `medium` resource level and algorithm strength of `4`. Compress files that are not modified in the last 5 days. Compress files with the name pattern *.arch only.

```
Storage> compress modage create tpcc_arch1 5
Storage> compress pattern create tpcc_arch1 \*.arch
Storage> compress schedule start tpcc_arch1 comp_sched_1 medium 4
```

List the scheduled compression job status for the tpcc_arch1 file system.

```
Storage> compress schedule list tpcc_arch1
Schedule Information for tpcc_arch1
============================================
Name            Node       Duration  Minute Hour   Day      Month    WeekDay
====            ====       ========  ====== ====   ===      =====    ========
comp_sched_1    any        3 hour(s)  0      1      *        *        6

Mod Age                 Algorithm      Resource        pattern
=======                 =========      ========        =======
1                       4              medium          *.arch
```

17.3.5 SEE ALSO

disk(1), hba(1), fencing(1), fs(1), pool(1), scanbus(1), snapshot(1), tier(1), iscsi(1), cifs(1)

17.4 fencing

17.4.1 SYNOPSIS

`fencing destroy`

`fencing on disk` [*disk1*, *disk2*, *disk3*]

`fencing on majority`

`fencing off`

`fencing replace` *src_disk* fIdest_disk

`fencing status`

17.4.2 DESCRIPTION

The storage `fencing` commands perform I/O fencing-related operations. These commands are not available in a single node set up.

17.4.3 OPTIONS

`fencing destroy` Destroy coordinator pool. This command has no effect for majority-based fencing.

`fencing on disk` [*disk1*, *disk2*, *disk3*] Enable disk-based I/O fencing on the three disks.

`fencing on majority` Enable majority-based I/O fencing.

`fencing off` Disable I/O fencing on all the nodes.

`fencing replace` *src_disk dest_disk* Replace an existing coordinator disk with another disk. If the disk being replaced is in a failed state, then it is mandatory to delete the disk from the array. This is required because if the failed disk comes up and works properly, it can lead to an even number of I/O fencing disks, and this affects the functionality. This command has no effect for majority-based fencing.

`fencing status` Check the I/O fencing status.

17.4.4 EXAMPLES

Destroy coordinator pool.

```
Storage> fencing destroy
```

Enable disk-based I/O fencing on the disks `ams_wms0_000d`, `ams_wms0_000e`, `ams_wms0_000f` (only three disks required).

```
Storage> fencing on disk ams_wms0_000d,ams_wms0_000e,ams_wms0_000f
ACCESS fencing Success V-288-0 IO Fencing feature now Enabled with SCSI3 Persistent␣
↪Reservations
100% [#] Enabling fencing
```

Check I/O fencing status after enabling I/O fencing.

```
Storage> fencing status

IO Fencing Status
=================
Enabled

Disk Name            Coord Flag On
==============       ==============
ams_wms0_000d        Yes
ams_wms0_000e        Yes
ams_wms0_000f        Yes
```

Disable I/O fencing on all the nodes.

```
Storage> fencing off
ACCESS fencing Success V-288-0 IO Fencing feature now Disabled
100% [#] Disabling fencing
```

Check I/O fencing status after disabling I/O fencing.

```
Storage> fencing status

IO Fencing Status
=================
Disabled

Disk Name            Coord Flag On
==============       ==============
ams_wms0_000d        Yes
ams_wms0_000e        Yes
ams_wms0_000f        Yes
```

Replace ams_wms0_000f with ams_wms0_000c with fencing in Enabled mode.

```
Storage> fencing replace ams_wms0_000f ams_wms0_000c
ACCESS fencing Success V-288-0 Replaced disk ams_wms0_000f with ams_wms0_000c␣
↪successfully.
100% [#] Replacing disk ams_wms0_000f with ams_wms0_000c
```

Replace ams_wms0_000f with ams_wms0_000c with fencing in Disabled mode.

```
Storage> fencing replace ams_wms0_000f ams_wms0_000c
ACCESS fencing Success V-288-0 Replaced disk ams_wms0_000f with ams_wms0_000c␣
↪successfully. Please run fencing on.
100% [#] Replacing disk ams_wms0_000f with ams_wms0_000c
```

Check I/O fencing status after I/O fencing replace operation completed.

```
Storage> fencing status

IO Fencing Status
=================
Disabled

Disk Name            Coord Flag On
==============       ==============
ams_wms0_000c        Yes
```

```
ams_wms0_000d       Yes
ams_wms0_000e       Yes
```

Enable majority-based I/O fencing (no disks required).

```
Storage> fencing on majority
ACCESS fencing WARNING V-288-0 The IO may be interrupted while switching fencing␣
↪state. Would you like to continue(yes/no):
yes
ACCESS fencing Success V-288-0 Majority Fencing Enabled
100% [#] Enabling fencing
```

Check I/O fencing status after enabling I/O fencing.

```
Storage> fencing status

IO Fencing Status
=================
Majority Fencing Enabled
```

Disable majority-based I/O fencing.

```
Storage> fencing off
ACCESS fencing WARNING V-288-0 The IO may be interrupted while switching fencing␣
↪state. Would you like to continue(yes/no):
yes
ACCESS fencing Success V-288-0 IO Fencing feature now Disabled
100% [#] Disabling fencing
```

Check fencing status after disabling fencing.

```
Storage> fencing status
IO Fencing Status
=================
Disabled
```

17.4.5 SEE ALSO

disk(1), hba(1), fs(1), pool(1), scanbus(1), snapshot(1), tier(1), iscsi(1), quota(1)

17.5 dedup

17.5.1 SYNOPSIS

`dedup enable` *fs_name* [*blksize*]

`dedup disable` *fs_name*

`dedup list` [*fs_name*]

`dedup start` *fs_name* [*node_name*]

`dedup stop` *fs_name*

`dedup status` [*fs_name*]

`dedup set cpu` *cpuvalue* [*fs_name*]

`dedup set memory` *memvalue*

`dedup set priority` *priorityval* [*fs_name*]

`dedup schedule set` *fs_name hours day* [*freq*]

`dedup schedule modify` *fs_name hours day* [*freq*]

`dedup schedule delete` *fs_name*

`dedup dryrun` *fs_name* [*threshold*]

`dedup remove` *fs_name*

17.5.2 DESCRIPTION

The `dedup` command performs administration functions for the Veritas Access deduplication feature. The `dedup` command exposes interfaces for configuring deduplication and performing different deduplication actions. This includes operations such as enabling, disabling, starting, stopping, and removing deduplication on a file system. The `dedup` command also enables you to reset several deduplication configuration parameters and display the current deduplication status. Some configuration parameters can be set as local (specific to the file system) and/or global (applicable to all deduplication-enabled file systems). The local parameter overrides the value of the global parameter.

17.5.3 OPTIONS

`dedup enable` *fs_name* [*blksize*] Enable deduplication on a file system. Deduplication must be enabled for a file system before setting its configuration parameters and schedule. This command will also re-enable a deduplication schedule for a file system. The *blksize* option should be specified in bytes, for example, 4096. It should be a power of 2, multiple of file system's block size and equal to or less than 128 K. Default block size is 0, which means that if the file system's block size is less than 4096 bytes, then the dedup block size is set to 4096 bytes. Otherwise the dedup block size is set to the same as the file system's block size.

`dedup disable` *fs_name* Disable deduplication schedule on a file system. Other information such as configuration, schedule, and the deduplication database remains intact.

Warning: Keeping deduplication disabled for longer periods of time may reduce the effectiveness of deduplication when it is re-enabled.

dedup list [*fs_name*] List the deduplication-configured file systems along with its granularity, state, schedule, and node list where it will run. Currently, node list is picked up automatically by the system and cannot be set by the user.

dedup start *fs_name* [*node_name*] Start deduplication manually. *fs_name* must be enabled for deduplication to start. You can specify the node for running the deduplication process.

dedup stop *fs_name* Stop the deduplication process running on a file system.

dedup status [*fs_name*] Display the current status of deduplication. It shows the file system name, saving's amount, status, node where deduplication is either running or has completed, deduplication type i.e., SCHEDULED or MANUAL and other details.

dedup set cpu *cpuvalue* [*fs_name*] Set the CPU limit. This parameter can be set as global and/or local. Possible *cpuval* are `idle` and `yield`.

dedup set memory *memvalue* Set the memory limit in MB. It should be in a range of 128-4096 MB, default value is 128 MB. This can be set only as a global parameter.

dedup set priority *priorityval* [*fs_name*] Set the priority for deduplication-enabled file system(s). This parameter can be set as global and/or a local. Possible *priorityval* are `low`, `normal`, and `high`.

dedup schedule set *fs_name hours day* [*freq*] Set the schedule for the deduplication-enabled file system. This can be set only as a local parameter. Two categories of schedule are allowed i.e., run periodicity and type periodicity. The granularity of the schedule is limited to time of day and day of month. The *hours* can be specified as * (every hour), * /N (every N hours) where N is in a range [1-12] or 5 comma-separated hours list in the range [0-23]. The *day* can be specified as * (every day), * /N (every N days) where N is in the range [1-15] or any number in the range [1-7], where 1 means Sunday, 2 means Monday, and so on. The *freq* can be specified in the range 1-5. Default value is 1. A value of 4 means every 4th run deduplicates the file system, whereas the other runs consolidate the changes.

dedup schedule modify *fs_name hours day* [*freq*] Modify the deduplication schedule. The *hours*, *day*, and *freq* values can be specified. See the `dedup schedule set` command for the values.

dedup schedule delete *fs_name* Delete the deduplication schedule.

dedup dryrun *fs_name* [*threshold*] Initiate a deduplication dry run on the file system *fs_name*. This reports the approximate space saving. Dry run can be converted to the actual deduplication if it meets the threshold value.

dedup remove *fs_name* Remove deduplication configuration related information from the specified file system *fs_name*. This command removes all the configurations and the database specific to that file system.

Warning: This operation cannot be undone and re-running deduplication may take significant time.

17.5.4 EXAMPLES

Enable deduplication on the file system `fs1`:

```
Storage> dedup enable fs1 blksize=8192
```

Set the deduplication schedule:

```
Storage> dedup schedule set fs1 0,6,12,18 2 3
```

List the deduplication-enabled file system(s):

```
Storage> dedup list
Default     Priority        CPU     Memory
-------------------------------------
NORMAL              IDLE    128M
Filesystem  Priority        CPU     Granularity     Enabled Schedule        NodeList
-----------------------------------------------------------------------------------------
↪---------
/vx/fs1             NORMAL  IDLE    8192B           YES             SET             ␣
↪node_01,node_02
```

Show the deduplication parameters details on file system *fs1*:

```
Storage> dedup list fs1
Priority                            NORMAL
CPU                                 IDLE
Memory                              128M
Granularity                 8192B
Enabled                             YES
Schedule hours                      Every hour
Schedule day interval       Every Sunday
Schedule frequency          1
NodeList                            node_01,node_02
```

Check whether the dedup dry run reaches the threshold value of 60%:

```
Storage> dedup dryrun fs1 60
```

Show the deduplication status:

```
Storage> dedup status
Filesystem          Saving  Status          Node            Type            Details
-----------------------------------------------------------------------------------------
↪--------
 /vx/fs1                    00%             COMPLETED               node_01 MANUAL ␣
↪2011/05/23 17:08:00 Dryrun
                                                    threshold not met. Cleaning up...
```

Start deduplication on all the file systems *fs1*:

```
Storage> dedup start fs1
```

Start deduplication on the file systems *fs1* on node *node_01*:

```
Storage> dedup start fs1 node_01
```

Show the deduplication status:

```
Storage> dedup status
Filesystem          Saving  Status          Node            Type            Details
-----------------------------------------------------------------------------------------
↪--------
 /vx/fs1                    --              RUNNING                 node_01 MANUAL  --
```

Show the status output after completion of deduplication:

```
Storage> dedup status
Filesystem          Saving  Status          Node            Type            Details
-----------------------------------------------------------------------------------------
↪--------
 /vx/fs1                    34%             COMPLETED               node_01 MANUAL ␣
↪2011/05/23 17:20:53 End detecting
                                                    duplicates and file system␣
↪changes 0
```

Remove deduplication on the file system *fs1*.:

```
Storage> dedup remove fs1
```

List deduplication-enabled file systems:

```
Storage> dedup list
Default  Priority  CPU    Memory
--------------------------------
         NORMAL    IDLE     128M
Filesystem                Priority  CPU    Granularity  Enabled  Schedule          ␣
↪NodeList
-----------------------------------------------------------------------------------------
↪---------
```

17.5.5 SEE ALSO

disk(1), hba(1), fencing(1), fs(1), pool(1), scanbus(1), snapshot(1), tier(1), iscsi(1), cifs(1)

17.6 disk

17.6.1 SYNOPSIS

`disk list` [stats | detail | paths]

`disk configure local` <*nodename*> <*vendor_id*> <*product_id*> [*serial_num*]

`disk grow` *disk_name*

`disk format` *disk1* [, *disk2*,...] [force]

`disk remove` *disk1* [, *disk2*,...]

The disks may be direct-attached storage or shared storage devices. The format command enables network sharing of local disks, cluster wide. Also to grow a selected disk, if it is resized on the storage array. With format command, you can format the disk to "simple" format, if the disk does not belong to any disk groups.

17.6.2 DESCRIPTION

The `disk list` command displays aggregated information of all the disk devices connected to any or all of the nodes in the cluster.

The `disk list stats` command displays a list of disks and nodes in a tabular form. Each row corresponds to a disk, and each column corresponds to a node. An "OK" in the table indicates that the disk that corresponds to that row is accessible by the node that corresponds to that column. An "ERR" indicates that the disk that corresponds to that row is inaccessible by the node that corresponds to that column. A "NOT_CONN" indicates that the disk that corresponds to that row is not connected to the node that corresponds to that column.

The `disk list detail` command displays a list of disks and their properties. The column `ID` in `disk list detail` consists of: `VendorID`, `ProductID`, `TargetID`, and `LunID`. A ":" separates these fields.

If a disk is shared across multiple nodes and has name inconsistencies across the nodes, then that disk is shown as "__".

The `disk list paths` command shows the list of multiple paths of disks connected to all or any of the nodes in the cluster. It also shows the status of those paths on each node in the cluster.

The `disk configure local` <*nodename*> <*vendor_id*> <*product_id*> [*serial_num*] command adds local disks having the specified vendor ID and product ID in the JBOD category. The command can be used only on local disks which have locally-unique disk UDIDs. Else, UDID conflicts with other node's local disks may occur. If you want to run this command on all the nodes, provide <nodename> argument as "all". The <*vendor_id*> and <*product_id*> can be derived from the UDID of a disk. UDID is composed of <*vendor_id*>, <*product_id*>, cabinet serial number, and LUN serial numbers joined by underscore(or %5F). If the <*vendor_id*> or <*product_id*> has a space (or %20) in it's name, replace it with '#' and provide as one-word argument in the command. It may happen that the default location to get the serial number is invalid and hence the serial number in UDID will be filled with "0000". This would still not make the UDID unique globally. `serial_num` should be given in the format `opcode/pagecode/offset/length` calculated carefully from the serial number format of the disk.

The `disk grow` *disk_name* increases the size of the specified disk. Before performing this operation, make sure that you increase the storage capacity of the disk on the storage array. **Caution:** When increasing the storage capacity of a disk, make sure that the storage array does not reformat it. This will destroy the data. For help, contact your Storage Administrator.

The `disk format` *disk1 [,disk2...]* [`force`] command formats the disk. If the disk does not belong to any disk group, the format erases the first 100M space on the disk, and the disk is formatted using the `vxdisksetup` command. Use the `force` option to force formatting of disks with invalid configurations.

The `disk remove` *disk1 [,disk2...]* command removes the disks from the cluster configuration. This command is helpful in cases when the disk attributes are incorrectly listed in the system. Only the disks that are not a part of the pools can be removed. The command does not destroy the data on the disk, but just removes the disk from the system configuration. Rebooting the system or running `scanbus` brings back the disk to the system's configuration. To remove the disk permanently from the system's configuration, the Storage Administrator should remove the disk's mapping from the array.

17.6.3 EXAMPLES

Display the list of disks in tabular form.

```
Storage> disk list stats
Disk              nasgw9_2  nasgw9_1  nasgw9_3  nasgw9_4  nasgw9_5  nasgw9_6
====              ======    ======    ======    ======    ======    ======
AMS_WMS0_0        OK        OK        OK        OK        OK        OK
AMS_WMS0_1        OK        OK        OK        OK        OK        OK
DS4800-0_0        OK        OK        OK        OK        OK        OK
DS4800-0_1        OK        OK        OK        OK        OK        OK
nasgw9_1_Disk_0   NOT_CONN  OK        NOT_CONN  NOT_CONN  NOT_CONN  NOT_CONN
nasgw9_1_Disk_1   OK        OK        OK        OK        OK        OK
EMC_CLARiiON0_0   OK        OK        OK        OK        OK        OK
EMC_CLARiiON0_1   OK        OK        OK        OK        OK        OK
```

Display the list of disks and their properties.

```
Storage> disk list detail
Disk            Pool              Enclosure   Array Type   Size (Use%)   Transport  ID ␣
↪                 Serial Number
====            ===============  ==========  ===========  ===========   ==========␣
↪==================  =============
ams_wms0_10     *coordinator*     ams_wms0    A/A          1.00G 0.0%     FC        ␣
↪HITACHI:DF600F:4:1  71011588000A
ams_wms0_11     *coordinator*     ams_wms0    A/A          1.00G 0.0%     FC        ␣
↪HITACHI:DF600F:4:2  71011588000B
ams_wms0_12     p03               ams_wms0    A/A          1.00G 27.5%    FC        ␣
↪HITACHI:DF600F:4:3  71011588000C
ams_wms0_13     p03               ams_wms0    A/A          1.00G 18.8%    FC        ␣
↪HITACHI:DF600F:4:4  71011588000D
ams_wms0_14     p04               ams_wms0    A/A          1.00G 17.8%    FC        ␣
↪HITACHI:DF600F:4:5  71011588000E
```

Show the list of multiple paths of disks connected to all or any of the nodes in the cluster. Also show the status of those paths on each node in the cluster.

```
Storage> disk list paths
Disk              Paths    nasgw78_2              nasgw78_1
====              =====    ========               ========
AMS_WMS0_0        Path 1   primary,enabled,active primary,enabled,active
AMS_WMS0_1        Path 1   primary,enabled,active primary,enabled,active
Disk_0            Path 1   enabled,active         enabled,active
nasgw78_2_Disk_0  Path 1   enabled,active         -
nasgw78_1_Disk_0  Path 1   -                      enabled,active
```

Add local disks having the specified vendor ID and product ID in the JBOD category. Before executing the command, note the ID.

```
Storage> disk list detail
Disk             Pool   Enclosure   Array Type   Size (Use%)   Transport   ID
↪                                                                          ␣
↪Serial Number
===============  =====  ==========  ===========  ============  ========== ␣
↪==========================================================================␣
↪ ================================
vassd_01_disk_0  -      disk        Disk         558.4G 0.0%   SCSI        DELL%5FPERC
↪%20H710%5FDISKS%5F6C81F660E7883400204FABF20704390E

Storage> disk configure local vassd_01 DELL PERC#H710 18/131/6/16
Please make sure that all disks from the given array are not in use.
Continue? (y/n, default:n): y
ACCESS Disk SUCCESS V-288-0 Configured local disks on the node vassd_01 successfully.
```

After the command is executed, the ID changes. Disk name may also change as a consequence.

```
Storage> disk list detail
Disk             Pool   Enclosure   Array Type   Size (Use%)   Transport   ID
↪                                                                          ␣
↪              Serial Number
===============  =====  ==========  ===========  ============  ========== ␣
↪==========================================================================
↪ ================================
vassd_01_disk_2  -      disk        Disk         558.4G 0.0%   SCSI        DELL%5FPERC
↪%20H710%5FDISKS%5F00106C81F660E7883400204FABF20704%5F%7B0ba66790-021b-11e7-a802-
↪80c008319b08%7D  00106C81F660E7883400204FABF20704

Storage> disk configure local all DELL PERC#H710
Please make sure that all disks from the given array are not in use.
Continue? (y/n, default:n): y
ACCESS Disk SUCCESS V-288-0 Configured local disks on all the nodes successfully.
```

Grow a selected disk.

```
Storage> disk grow Disk_0
ACCESS disk SUCCESS V-288-0 disk grow Disk_0 completed successfully.
```

Format a specified disk.

```
Storage> disk format aluadisk0_6
You may lose all the data on the disk, do you want to continue (y/n, the default is␣
↪n):y
disk format: aluadisk0_6 has been formatted successfully.

Storage> disk format aluadisk0_7, aluadisk0_8
You may lose all the data on the disk, do you want to continue (y/n, the default is␣
↪n):y
ACCESS Disk SUCCESS V-288-0 disk format: aluadisk0_7 has been formatted successfully.

ACCESS Disk ERROR V-288-0 Disk aluadisk0_8 has invalid pool configuration. Format␣
↪with force option to clean up else run scanbus to retrieve.
Storage> disk format aluadisk0_8 force
You may lose all the data on the disk, do you want to continue (y/n, the default is␣
↪n):y
ACCESS Disk SUCCESS V-288-0 disk format: aluadisk0_8 has been formatted successfully.

Storage> disk format aluadisk0_7
```

```
You may lose all the data on the disk, do you want to continue (y/n, the default is␣
↪n):y
ACCESS disk ERROR V-288-832 Disk aluadisk0_7 is used by pool pool_1, remove the disk␣
↪from the use by command  pool rmdisk  firstly and then format.
ACCESS disk INFO V-288-832  File System fs_1 is based on the disk aluadisk0_7.
ACCESS disk INFO V-288-832  File System fs_2 is based on the disk aluadisk0_7.
ACCESS disk INFO V-288-832  File System fs_4 is based on the disk aluadisk0_7.
```

Remove a specified disk from configuration.

```
Storage> disk remove emc0_03ff
Removed disk emc0_03ff from nasgw78_1
Removed disk emc0_03ff from nasgw78_2
```

17.6.4 SEE ALSO

hba(1), fencing(1), fs(1), pool(1), scanbus(1), snapshot(1), tier(1), iscsi(1), quota(1)

17.7 fs

17.7.1 SYNOPSIS

`fs create simple` *fs_name size pool1* [, *disk1*,...] [blksize=<bytes>] [pdir_enable={yes|no}] [encrypt={on|off}] [worm={yes|no}]

`fs create mirrored` *fs_name size nmirrors pool1* [, *disk1*,...] [protection=disk|pool]
[blksize=<bytes>] [pdir_enable={yes|no}] [encrypt={on|off}] [worm={yes|no}]

`fs create striped` *fs_name size ncolumns pool1* [, *disk1*,...]
[stripeunit=<kilobytes>] [blksize=<bytes>] [pdir_enable={yes|no}] [encrypt={on|off}] [worm={yes|no}]

`fs create mirrored-stripe` *fs_name size nmirrors ncolumns pool1*
[, *disk1*,...] [protection=disk|pool] [stripeunit=<kilobytes>] [blksize=<bytes>] [pdir_enable={yes|no}] [encrypt={on|off}] [worm={yes|no}]

`fs create striped-mirror` *fs_name size nmirrors ncolumns pool1*
[, *disk1*,...] [protection=disk|pool] [stripeunit=<kilobytes>] [blksize=<bytes>] [pdir_enable={yes|no}] [encrypt={on|off}] [worm={yes|no}]

`fs create ecoded` *fs_name size ncolumns nparity pool1*
[, *disk1*,...] [blksize=<bytes>] [stripeunit=<kilobytes>] [stripe_aligned={yes|no}] [stripe_tag={node|disk}] [rotating_parity={yes|no}] [workload={virtualmachine|mediaserver}]

`fs create largefs simple` *fs_name size pool1* [*blksize=<bytes>*]

`fs create largefs mirrored` *fs_name size nmirrors pool1* [blksize=<bytes>]

`fs create largefs striped` *fs_name size ncolumns pool1* [stripeunit=<kilobytes>] [blksize=<bytes>]

`fs create largefs mirrored-stripe` *fs_name size nmirrors ncolumns pool1*
[stripeunit=<kilobytes>] [blksize=<bytes>]

`fs create largefs striped-mirror` *fs_name size nmirrors ncolumns pool1*
[stripeunit=<kilobytes>] [blksize=<bytes>]

`fs create pretuned` *fs_name size pool1* workload={virtualmachine|mediaserver}
[layout={striped|striped-mirror|mirrored-stripe}] [ncolumns] [encrypt={on|off}]

`fs defrag now` *fs_name time*

`fs destroy` *fs_name*

`fs list` [*fs_name*]

`fs online` *fs_name*

`fs offline` *fs_name*

`fs growto` primary|secondary *fs_name new_length* [*pool1* [, *disk1*,... [protection=disk|pool]

`fs growby` primary|secondary *fs_name length_change* [*pool1* [, *disk1*,...]] [protection=disk|pool]

`fs shrinkto` primary|secondary *fs_name new_length*

`fs shrinkby` primary|secondary *fs_name length_change*

`fs addmirror` *fs_name pool1* [, *disk1*,...] [protection=disk|pool]

`fs rmmirror` *fs_name [pool_or_disk_name]*

`fs addcolumn` *fs_name ncolumns pool_or_disk_name*

`fs rmcolumn` *fs_name*

`tier addcolumn` *fs_name ncolumns pool_or_disk_name*

`tier rmcolumn` *fs_name*

`fs checkmirror`

`fs resync` [*fs_name*]

`fs checkresync`

`fs setfastresync` *fs_name [pool_or_disk_name]*

`fs unsetfastresync` *fs_name*

`fs fsck` *fs_name*

`fs alert set` numinodes|numspace|fullspace value [*fs_name*,...] [*snapshot_name*]

`fs alert unset` numinodes|numspace|fullspace [*fs_name*,...] [*snapshot_name*]

`fs alert show`

`fs upgrade` *fs_name*

`fs defrag schedule create` *sched_name duration minute [hour] [day] [month] [day_of_week] [node]*

`fs defrag schedule remove` *sched_name*

`fs defrag schedule show` *[sched_name]*

`fs defrag schedule start` *fs_name sched_name*

`fs defrag schedule stop` *fs_name*

`fs defrag schedule list` *fs_name*

`fs policy add operation=move` *policy_name fs_name from_tier to_tier*
retrieval_option=Expedited|Standard|Bulk pattern [`atime` *condition*] [`mtime` *condition*]

`fs policy add operation=delete` *policy_name fs_name from_tier*
pattern [`atime` *condition*] [`mtime` *condition*]

`fs policy modify` *policy_name pattern* [`atime` *condition*] [`mtime` *condition*]

`fs policy delete` *policy_name fs_name*

`fs policy rename` *old_policy_name new_policy_name*

`fs policy list` *[fs_name]*

`fs policy run` *policy_name*

`fs policy dryrun` *policy_name*

`fs policy status` *policy_name*

`fs policy abort` *policy_name*

`fs policy pause` *policy_name*

`fs policy resume` *policy_name*

`fs policy schedule create` *fs_name [minute] [hour] [day_of_the_month] [month] [day_of_the_week]*

`fs policy schedule modify` *fs_name [minute] [hour] [day_of_the_month] [month] [day_of_the_week]*

`fs policy schedule remove` *fs_name*

`fs policy schedule list` *fs_name*

`fs retention set` *path rtime*

`fs retention show` *path*

`fs retention clear` *path*

`fs worm set` *fs_name*

`fs worm clear` *fs_name*

‘‘fs setmaxiops ‘‘ [*fs_name*] *maxiops*

`fs maxiopslist` [*fs_name*]

`fs iopsstat list` [*fs_name*]

`fs iopsstat reset` [*fs_name*]

17.7.2 DESCRIPTION

The storage `fs` commands manage file system operations.

File systems consist of metadata and file data. Metadata consists of information like last modification time, creation time, permissions, and so on. The total amount of space taken by the metadata depends upon the number of files. A file system that contains many small files requires more space to store the metadata, and a file system with fewer large files requires less space for storing the metadata.

When a file system is created, some space is set aside initially for the metadata. This space is generally proportional to the size of the file system. This is the reason for the non-zero usage percentage in the output of `fs list` just after the creation of the file system. The space set aside for the metadata grows or shrinks as and when required. A file system on a 1 GB volume takes around 40 MB (about 4%) initially for storing the metadata, whereas a file system of size 10 MB takes around 7.3 MB (73%) initially for storing the metadata.

Veritas Access provides a scale-out file system that manages a single namespace spanning over both on-premises storage as well as cloud storage, which provides better fault tolerance for large data sets.

Veritas Access supports access to a scale-out file systems using NFS-Ganesha, S3, CIFS and FTP protocols. You can use any Virtual IP to access data over these protocols.

S3 buckets created on scale-out file system must be accessed using a virtual-hosted-style URL (rather than the path-style URL) and S3 client's DNS must be updated to this virtual IP address for the corresponding virtual-hosted-style URL. If a bucket `bucket1` is created by the S3 client, then its virtual-hosted-style URL would be `bucket1.s3.cluster_name:8143`, where `cluster_name` is the Veritas Access cluster name and 8143 is the port on which the Veritas Access S3 server is running.

A scale-out file system is structured as a layered file system that includes a set of storage containers. The data that is stored in the cloud can be one of the storage containers. One on-premises storage container stores the metadata and all the other containers (on-premises and cloud) store the actual data. This modular structure allows the scale-out file system to be more resilient in cases where high capacity or fault tolerance is needed. A scale-out file system accomplishes this without compromising on file system performance.

Scale-out file system specifications

- Twenty percent of a scale-out file system's size is devoted to the metadata container.
- The maximum size of a metadata container is 10 TB.
- The minimum size of a scale-out file system is 10 GB.
- The maximum size of a scale-out file system is 3 PB.
- You can grow the scale-out file system up to 3 PB.
- To create or grow a scale-out file system above 522 TB, you need to provide the file system size in multiples of 128 GB.
- You can shrink the scale-out file system only if its size is less than 522 TB.

New storage containers are created when you grow the scale-out file system beyond 522 TB. The pool on which the scale-out fs is created will be used to create these new containers. There will also be a data movement to these new storage containers so that data is distributed evenly among all the storage continers(on-premises).

The following types of clouds can be added as storage containers for scale-out fs, Amazon S3, Amazon Glacier, AWS GovCloud(US), Azure, Google cloud, Alibaba, Veritas Access S3 and any S3-compatible storage provider. The data is always written to the on-premises storage container and then data can be moved to the cloud container using a tiering mechanism. File metadata including any attributes set on the file resides on-premises eventhough file is moved to cloud. This cloud as a tier feature is best used for moving infrequently accessed data to the cloud.

Amazon Glacier is an offline cloud tier, which means that data moved to Amazon Glacier cannot be accessed immediately. An EIO error is returned if you try to read, write, or truncate the files moved to the Amazon Glacier tier. If you want to read or modify the data, move the data to on-premises using `tier move` or using policies. The data will be available after some time based on the Amazon Glacier retrieval option you selected.

When Amazon S3, AWS GovCloud(US), Azure, Google cloud, Alibaba, Veritas Access S3 and any S3-compatible storage providere is used as the cloud tier, the data present on these clouds can be accessed any time (Unlike in Amazon Glacier). An EIO error is returned if you try to write, or truncate the files moved to these clouds. If you want to modify the data, move the data to on-premises using `tier move` or using policies.

Note: Not all of the storage commands are supported for scale-out file systems.

17.7.3 OPTIONS

size Size of the file system (for example 10m, 10M, 25g, 100G).

nmirrors Number of mirrors.

ncolumns Number of columns.

protection=disk If the protection is set to `disk`, then mirrors are created on separate disks. This flag works only for the file systems of type mirrored, mirrored-striped, and striped-mirror. The disks may or may not be in the same pool.

protection=pool If the protection is set to `pool`, then mirrors are created in separate pools. This flag works only for the file systems of type mirrored, mirrored-striped, and striped-mirror. If not enough space is available, then the file system creation operation fails. The `protection=pool` option is not supported for an isolated pool. If the file system is created with the `protection=pool` option, then both pools in question cannot be merged to create a single pool, which defeats the purpose of `protection=pool`.

stripeunit=<kilobytes> Set the stripe width of the file system, where possible values of kilobytes are 128, 256, 512, 1024, and 2048.

blksize=<bytes> Set the block size of the file system, where possible values of bytes are 1024, 2048, 4096, and 8192. Default block size is 8192.

pdir_enable={yes|no} Enable or disable partition directory for the file system. The default value is *pdir_enable=no*, which means partition directory is disabled for the file system.

encrypt={yes|no} Specify whether to create the file system on encrypted volume. If set to `on`, file system is created on encrypted volume. The default value is *encrypt=off*.

worm={yes|no} Specify whether to create a worm-enabled file system. If set to `yes`, file system is created with WORM support enabled. The default value is *worm=no*.

workload={virtualmachine|mediaserver} Type of workload using this file system. The workload type cannot be changed after the file system is created. If the workload type is virtualmachine, then all the files in the file system have a minimum extent size of 1MB. This reduces the file system fragmentation and improves virtual machine I/O performance.

layout={striped|striped-mirror|mirrored-stripe} Layout of the underlying volume on which the file system is created. The default value is layout=striped.

17.7.4 COMMAND DESCRIPTIONS

`fs create simple` *fs_name size pool1* [, *disk1*,...] [blksize=<bytes> [pdir_enable={yes|no}] [encrypt={on|off}] [worm={yes|no}]

Create a simple file system with the specified size on one of the specified pools/disks and make it online.

`fs create mirrored` *fs_name size nmirrors pool1* [, *disk1*,...] [protection=disk|pool] [blksize=<bytes>] [pdir_enable={yes|no}] [encrypt={on|off}] [worm={yes|no}]

Create a mirrored file system with the specified number of mirrors and make it online.

`fs create striped` *fs_name size ncolumns pool1* [, *disk1*,...] [stripeunit=<kilobytes>] [blksize=<bytes>] [pdir_enable={yes|no}] [encrypt=on|off}] [worm={yes|no}]

Create a striped file system with the specified number of stripes and make it online.

`fs create mirrored-stripe` *fs_name size nmirrors ncolumns pool1* [, *disk1*,...] [protection=disk|pool] [stripeunit=<kilobytes>] [blksize=<bytes>] [pdir_enable={yes|no}] [encrypt={on|off}] [worm={yes|no}]

Create a mirrored-striped file system with a specified number of mirrors and stripes and make it online.

`fs create striped-mirror` *fs_name size nmirrors ncolumns pool1* [, *disk1*,...] [protection=disk|pool] [stripeunit=<kilobytes>] [blksize=<bytes>] [pdir_enable={yes|no}] [encrypt={on|off}] [worm={yes|no}]

Create a striped-mirrored file system with a specified number of mirrors and stripes and make it online.

`fs create ecoded` *fs_name size ncolumns nparity pool1* [, *disk1*,...] [blksize=<bytes>] [stripeunit=<kilobytes>] [stripe_aligned={yes|no}] [stripe_tag={node|disk}] [rotating_parity={yes|no}] [workload={virtualmachine|mediaserver}]

Create an ecoded file system with a specified number of data columns and parity columns and make it online.

`fs create largefs simple` *fs_name size pool1* [blksize=<bytes>]

Create a largefs file system with the specified size on pools/disks specified and bring it online.

`fs create largefs mirrored` *fs_name size nmirrors pool1* [blksize=<bytes>]

Create a mirrored largefs file system with the specified number of mirrors and make it online.

`fs create largefs striped` *fs_name size ncolumns pool1* [stripeunit=<kilobytes>] [blksize=<bytes>]

Create a striped largefs file system with the specified number of stripes and make it online.

`fs create largefs mirrored-stripe` *fs_name size nmirrors ncolumns pool1* [stripeunit=<kilobytes>] [blksize=<bytes>]

Create a mirrored-striped largefs file system with a specified number of mirrors and stripes and make it online.

`fs create largefs striped-mirror` *fs_name size nmirrors ncolumns pool1* [stripeunit=<kilobytes>] [blksize=<bytes>]

Create a striped-mirrored largefs file system with a specified number of mirrors and stripes and make it online.

`fs create pretuned` *fs_name size pool1* workload={virtualmachine|mediaserver} [layout={striped|striped-mirror|mirrored-stripe}] [ncolumns] [encrypt={on|off}]

Create a file system with the specified size using the disks from the specified pool. The volume is created with the specified layout. Stripe the volume across the specified number of columns with 512 K as the stripe size. Tune the file system for the specified workload.

`fs defrag now` *fs_name time*

Defragment a file system now. The *time* value should be larger than 1 minute or infinite.

`fs destroy` *fs_name*

Destroy a file system.

`fs list` [*fs_name*]

List all the file systems.

`fs online` *fs_name*

Make the file system online.

`fs offline` *fs_name*

Make the file system offline.

`fs growto` primary|secondary *fs_name new_length* [*pool1* [, *disk1*,...]] [protection=disk|pool]

Grow the file system's primary or secondary tier to a specified size. If no pool is specified with the command, the disks for growing the file system can be taken from any available pool. The protection flag takes the default value of "disk" in this case. The value of protection field cannot be set to "pool" when no pool is specified with the command. This operation may convert the layout of the file system if the command determines that the new file system is too large for the original layout.

If the file system specified is a scale-out file system (`largefs` type), growing the primary tier grows the default disk tier. Growing the secondary tier is not supported for a scale-out file system.

`fs growby` primary|secondary *fs_name length_change* [*pool1* [, *disk1*,...]] [protection=disk|pool]

Grow the file system's primary or secondary tier by a specified size. If no pool is specified with the command, the disks for growing the file system can be taken from any available pool. The protection flag takes the default value of `disk` in this case. The value of protection field cannot be set to `pool` when no pool is specified with the command. This operation may convert the layout of the file system if the command determines that the new file system is too large for the original layout. See the description of `fs growto` for more information if you resize a scale-out file system.

`fs shrinkto` primary|secondary *fs_name new_length*

Shrink the file system's primary or secondary tier to a specified size.

See the description of `fs growto` for more information if you resize a scale-out file system.

`fs shrinkby` primary|secondary *fs_name length_change*

Shrink the file system's primary or secondary tier by a specified size.

See the description of `fs growto` for more information if you resize a scale-out file system.

`fs addmirror` *fs_name pool1* [, *disk1*,...] [protection=disk|pool]

Add a mirror to the file system.

`fs rmmirror` *fs_name [pool_or_disk_name]*

Remove a mirror from the file system that is spanning on the specified pools/disks. If a pool name is the same as a disk name, then the mirror present on the pool is deleted.

`fs addcolumn` *fs_name ncolumns pool_or_disk_name*

Add the specified number of columns to the file system. In case of a striped file system, the number of disks specified should be equal to *ncolumns*. In case of mirrored-stripe and striped-mirror, the disks specified should be eqaul to (ncolumns * number_of_mirrors_in_fs).

`fs rmcolumn` *fs_name*

Remove a column from the file system.

`tier addcolumn` *fs_name ncolumns pool_or_disk_name*

Add specified number of columns to the secondary tier of the file system. In case of a striped file system, the number of disks specified should be equal to *ncolumns*. In case of mirrored-stripe and striped-mirror, the disks specified should be equal to (ncolumns * number_of_mirrors_in_fs).

`tier rmcolumn` *fs_name*

Remove a column from the secondary tier of a file system.

`fs checkmirror`

Show file systems that have stale mirrors.

`fs resync` [*fs_name*]

Resynchronize all stale mirrors for all file systems or for a certain file system.

`fs checkresync`

Show resynchronization and rollback synchronization progress running in the background.

`fs setfastresync` *fs_name [pool_or_disk_name]*

Enable fast resync for the specified file system.

`fs unsetfastresync` *fs_name*

Disable fast resync for the specified file system.

`fs fsck` *fs_name*

Check and repair the specified file system.

Scale-out fsck checks the consistency of the metafs, the datafs, and the database and repairs any inconsistencies found. Scale-out fsck checks if the metafs and datafs are marked for fullfsck, and if yes, it performs a fullfsck (check and repair) of the corresponding file systems. Based on the actions taken by fsck on the individual file systems, Scale-out fsck repairs the inconsistencies in other parts of the scale-out file system. Scale-out fsck goes through all the file handles present in the database and checks if the corresponding metafs and datafs file handles are consistent with each other. For example, in some cases, fullfsck might delete files from the datafs. To maintain consistency, corresponding metafs files and other datafs files are removed, and the corresponding key is removed from the database.

`fs alert set numinodes|numspace|fullspace value` *[fs_name]* [*snapshot_name*]

Allow users to set alerts on file systems and snapshot usage. The user can set alerts based on the number of inodes used or the file system space used. The value should be in a percentage when the alert is being set for the disk space. The default alert set for the disk space usage is at 80%. When the alert set is numinodes, the input should be the number of inodes. The default alert value for numinodes is set at 0. This will not send any alert till you set it to a different value. When this command is not supplied with the file system name, it changes the default system wide value for the alerts. For the file systems on which alerts are set explicitly, the alert is sent based on the value specified and not the default value. The alerts can be observed by the user in the "Report> showevents" CLI command. You can specify a comma-separated list of file systems.

fullspace is the tunable for file system full protection. For the file system to run efficiently, users should always reserve some space for the file system, instead of using 100% of the space. When file system usage is above the limit set by fullspace, all the NFS/CIFS shares on top of the file system are automatically changed to readonly to prevent the file system from becoming full. When users grow the file system or delete some files to get enough free space, the shares are automatically changed back to readwrite (there might be a delay for up to 5 minutes). By default, fullspace is 0, which means this function is disabled.

`fs alert unset numinodes|numspace|fullspace` *[fs_name]* [*snapshot_name*]

Allow users to unset the alerts set on the file system and the snapshot name. If an alert on any file system is unset, the user gets an alert for the file system based on the default values. You can specify a comma-separated list of file systems. When fullspace is unset (or set to 0), the shares that were changed to readonly due to file system high usage are changed back to readwrite mode immediately.

`fs alert show`

Show the current disk space usage and the alert value set. "(D)" beside the value shows that the value is the default value through the system.

`fs upgrade` *fs_name*

This command upgrades the file system layout to version 13. The file system needs to be online for this command to proceed. The version information about the file system is displayed in the `fs list fs_name` command.

`fs defrag schedule create` *sched_name duration minute [hour] [day] [month] [day_of_week] [node]*

Create a defrag schedule. The time and frequency of this schedule are specified in the command line in crontab format. This scheduled job runs on the node specified. If this node is not online at the time of the job, then this particular defrag job runs on the CFS primary. The scheduled defrag job may last up to the specified duration hours or minutes.

`fs defrag schedule remove` *sched_name*

Remove the defrag schedule by name *sched_name*. Make sure none of the file systems have this *sched_name* assigned.

`fs defrag schedule show` *[sched_name]*

Show the defrag schedule information for the schedule by name *sched_name*. If *sched_name* is not specified, all the information about all the schedules are displayed.

`fs defrag schedule start` *fs_name sched_name*

Assign or start the defrag schedule by name *sched_name* for the file system by name *fs_name*.

`fs defrag schedule stop` *fs_name*

Stop the scheduled defrag schedule for the file system by name *fs_name*.

`fs defrag schedule list` *fs_name*

List the scheduled defrag job details for the file system by name *fs_name*.

`fs policy add operation=move` *policy_name fs_name from_tier to_tier retrieval_option=Expedited\Standard\Bulk pattern* [`atime` *condition*] [`mtime` *condition*]

Create a data movement policy for the scale-out file system *fs_name*. The policy moves files and directories that meet the criteria from the source tier (*from_tier*) to the destination tier (*to_tier*). Movement can be from the cloud tier to a disk tier or from a disk tier to the cloud tier.

The retrieval option determines the time needed to move files from Amazon Glacier to on-premises. This option is not used when moving files from on-premises to Amazon Glacier.

Expedited retrievals typically complete within 1-5 minutes. The expedited option is expensive and you should use it conservatively. Files moved from the Amazon Glacier tier with the expedited option might return with the following error: InsufficientCapacityException (503 service unavailable). This error occurs if there is insufficient capacity to process the expedited request. This error only applies to expedited retrievals and not to standard or bulk retrievals.

Standard retrievals typically complete within 3-5 hours.

Bulk retrievals typically complete within 5-12 hours. Bulk is the default option.

Note: The maximum file size for moving files to AWS Glacier is 4 GB.

Pattern identifies the files or directories that you want to move between tiers. Pattern is required. You can further restrict the files or directories to move by specifying the last accessed time (`atime`) or the last modified time (`mtime`). The `atime` and `mtime` criteria are optional.

`fs policy add operation=delete` *policy_name fs_name from_tier pattern* [`atime` *condition*] [`mtime` *condition*]

Create a data deletion policy for the scale-out file system *fs_name*. The policy deletes files and directories that meet the criteria from the specified tier (*from_tier*). Pattern identifies the files or directories that you want to delete. Pattern is required. You can further restrict the files or directories to delete by specifying the last accessed time (`atime`) or the last modified time (`mtime`). The `atime` and `mtime` criteria are optional.

`fs policy modify` *policy_name pattern* [`atime` *condition*] [`mtime` *condition*]

Modifies the pattern, atime, and mtime search criteria for the file movement or deletion of the policy. atime and mtime are optional criteria. You cannot change the policy operation, or change the storage tiers that were specified when the policy was created.

`fs policy delete` *policy_name fs_name*

Stop any data movement or data deletion policy that was set for a file system. You cannot delete a policy if the policy is running.

`fs policy rename` *old_policy_name new_policy_name*

Rename an existing policy to a new policy name. You cannot rename a policy if the policy is running.

`fs policy list [fs_name]`

List all the data movement and data deletion polices that are set for all the file systems. If *fs_name* is included in the command, then the command lists all the policies for the specified file system.

`fs policy run` *policy_name*

Move or prune files according to the configured file system policy *policy_name*. The policy runs in the background until it gets completed, or until you abort or pause the policy. Filesystem policies are configured only for scale-out file systems (largefs type). If a policy run encounters an error, it is retried 5 times before the run is aborted.If the policy is already active but paused, you are prompted on whether you want to resume the paused job or you want to start a new run. You cannot run a policy if poilcy is scheduled or the last instance is still running. You also cannot run a policy if a tier move command is running for the same path or file system.

`fs policy dryrun` *policy_name*

Perform a dry run of the file system policy *policy_name*. This command previews the running of the policy but does not actually move or prune any files. Use this command to estimate the I/O activity involved if the policy is run. The dry run collects the statistics such as the number of files and the amount of data that will be moved. You cannot perform a dryrun if the policy is already running, or if a `tier move` command is running on the same path or file system.

`fs policy status` *policy_name*

Show the status of the currently running policy or dry run of the policy *policy_name*. If the policy is not currently running, the status shows the most recent run. The information includes the type of the run (normal or dry run), the status, the total data, the amount of data moved so far, the number of files, the number of files moved or deleted so far, and the name of the last file completed by the policy run.

`fs policy abort` *policy_name*

Aborts the currently executing policy run or dry run for the policy *policy_name*. The abort is immediate and interrupts any data movement in progress. The aborted policy cannot be resumed using the `fs policy resume` command. If you need to restart the policy, start a new run with the `fs policy run` command.

`fs policy pause` *policy_name*

Pause the currently executing policy run or dry run for the policy *policy_name*. The pause is immediate and interrupts any data movement in progress. You can resume the policy using the `fs policy resume` command. You can abort the paused policy using the `fs policy abort` command. When you pause a policy, the scheduled runs of the policy are also skipped until the policy is resumed.

`fs policy resume` *policy_name*

Restart the policy run or dry run that was paused with the `fs policy pause` command. The policy resumes from the point where the policy run was interrupted.

`fs policy schedule create` *fs_name [minute] [hour] [day_of_the_month] [month] [day_of_the_week]*

Create the file system policy schedule for the file system *fs_name*. The schedule uses a time format similar to the format used in UNIX cron configuration files.

The schedule applies to the file system policies created using the `fs policy create` command. When a schedule is set for a particular file system, all the policies for that file system are started at the scheduled times. If any policy is paused using the `fs policy pause` command, the policy does not run at the scheduled time. The policy is skipped until the policy is resumed using the `fs policy resume` command.

About the schedule format:

A schedule is specified in a format similar to the UNIX crontab format. The format uses five fields to specify when the schedule runs:

`minute` Enter a numeric value between 0-59, or an asterisk (*), which represents every minute. You can also enter a step value (*/x), or a range of numbers separated by a hyphen.

`hour` Enter a numeric value between 0-23, or an asterisk (*), which represents every hour. You can also enter a step value (*/x), or a range of numbers separated by a hyphen.

`day_of_the_month` Enter a numeric value between 1-31, or an asterisk (*), which represents every day of the month. You can also enter a step value (*/x), or a range of numbers separated by a hyphen.

`month` Enter a numeric value between 1-12, or an asterisk (*), which represents every month. You can also use the names of the month. Enter the first three letters of the month (you must use lowercase letters). You can also enter a step value (*/x), or a range.

`day_of_the_week` Enter a numeric value between 0-6, where 0 represents Sunday, or an asterisk (*), which represents every day of the week. You can also enter the first three letters of the week (you must use lowercase letters). You can also enter a step value (*/x), or a range.

A step value (*/x) specifies that the schedule runs at an interval of x. The interval should be an even multiple of the field's range. For example, you could specify */4 for the hour field to specify every four hours, since 24 is evenly divisible by 4. However, if you specify */15, you may get undesired results, since 24 is not evenly divisible by 15. The schedule runs after 15 hours, then 7 hours.

A range of numbers (two values separated by a hyphen) represents a time period during which you want the schedule to run.

Examples:

To run the schedule every two hours every day: 0 */2 * * *

To run the schedule on 2:00 a.m. every Monday: *2 * * 1

To run the schedule at 11:15 p.m. every Saturday: 15 23 * * 6

`fs policy schedule modify` ***fs_name [minute] [hour] [day_of_the_month] [month] [day_of_the_week]*** Modify the file system policy schedule for the file system *fs_name*. The schedule uses a time format similar to the format used in UNIX cron configuration files. Refer to the `fs policy schedule create` section for the detailed description of the schedule format.

`fs policy schedule remove` ***fs_name*** Remove the file system policy schedule associated with the file system *fs_name*.

`fs policy schedule list` ***fs_name*** List the file system policy schedule associated with the file system *fs_name*.

`fs retention set` ***path rtime*** Sets the retention on a file path *path* or on all the files that are currently present in specified path *path*. *rtime* can be in [1-9](d|D|m|M|y|Y) or mm-dd-yyyy format.

`fs retention show` ***path*** Shows the retention value applied on the specified file path *path*.

`fs retention clear` ***path*** Clears the retention on a file path *path* or on all the files that are currently present in specified path *path*.

`fs worm set` ***fs_name*** Enable worm support for the specified file system.

`fs worm clear` ***fs_name*** Disable worm support for the specified file system.

''fs setmaxiops '' [*fs_name*] *maxiops*

set/reset MAXIOPS on a file system

`fs maxiopslist`

List the value of maximum I/O operations per second (MAXIOPS) set on all the file systems.

`fs iopsstat list` [*fs_name*]

Display a statistical view of the maximum I/O operations per second (MAXIOPS) for a file system.

`fs iopsstat reset` [*fs_name*]

Reset the MAXIOPS satisticis instead of printing them.

17.7.5 EXAMPLES

Create a mirrored file system with name `fs1` with disks from the pools `pool1` and `pool2`.

```
Storage> fs create mirrored fs1 100M 2 pool1,pool2
100% [#] Creating mirrored filesystem
```

Add a mirror to file system fs1.

```
Storage> fs addmirror fs1 pool3,pool4
```

Remove a mirror from file system fs1 residing on Disk AMS_WMS0_0.

```
Storage> fs rmmirror fs1 AMS_WMS0_0
```

Add two columns to file system fs1.

```
Storage> fs addcolumn fs1 2 pool3
```

Remove column from file system fs1.

```
Storage> fs rmcolumn fs1
```

Add two columns to the secondary tier of file system fs1.

```
Storage> tier addcolumn fs1 2 pool3
```

Remove column from the secondary tier of file system fs1.

```
Storage> tier rmcolumn fs1
```

Show file systems that have stale mirrors.

```
Storage> fs checkmirror
fs_name:
----------
mirror3
```

Resynchronize all stale mirrors for all file systems.

```
Storage> fs resync
Resync stale mirrors for file systems that are started in the background.
```

Resynchronize all stale mirrors for a certain file system.

```
Storage> fs resync mirror3
Resync stale mirror for file system mirror3 is started in background.
```

Show resynchronization progress running in the background.

```
Storage> fs checkresync
FS                  MIRROR              TYPE      PROGRESS  START_TIME                ␣
↪  USED_TIME   REMAINING_TIME
==========================================================================================
mir2                tier 1,mirror 02    RESYNC    6.46%     Jun/05/2011/09:39:53      ␣
↪0:5:9       1:14:34
mir3_roll2          tier 1              ROLLBACK  1.28%     Jun/05/2011/14:51:40      ␣
↪0:0:12      15:23
mir3                tier 1,mirror 03    RESYNC    7.67%     Jun/05/2011/15:10:26      ␣
↪0:1:14      14:50
```

Make a file system offline.

```
Storage> fs offline fs1
100% [#] Offline filesystem
```

Check and repair a file system while it is in offline state. It tries normal fsck first, if fullfsck flag is set, it proceeds further accordingly based on the given input.

```
Storage> fs fsck fs1
Do you want to do fsck without log replay? yes|no
n
File system fs1 fsck successfully
Storage>
Storage> fs fsck fs1
Do you want to do fsck without log replay? yes|no
y
File system fs1 fsck successfully
Storage>
Storage> fs fsck fs1
Do you want to do fsck without log replay? yes|no
n
File system fs1 marked for full fsck.Running full fsck may take long time for␣
↪completion.
Do you want to continue? yes|no
n
File system fs1 full fsck canceled.
Storage>
Storage> fs fsck fs1
Do you want to do fsck without log replay? yes|no
n
File system fs1 marked for full fsck.Running full fsck may take long time for␣
↪completion.
Do you want to continue? yes|no
y
Storage>
Check the fullfsck status by running the command again.
Storage>
Storage> fs fsck fs1
fsck of largefs fs1 is successful
Storage> fs list mir3
General Info:
===============
```

```
Block Size:       1024 Bytes
Version:          Version 11
Cluster5_01:      online

Primary Tier
============
Size:            30.00G
Use%:            0%
Layout:          mirrored
Mirrors:         4
Columns:         -
Stripe Unit:     0.00 K
FastResync:      Enabled

1. Mirror 01:
List of pools:   pool1
List of disks:   disk1 disk2

2. Mirror 02:
List of pools:   pool1
List of disks:   disk3 disk4

3. Mirror 03:
List of pools:   pool1
List of disks:   disk5 disk6

4. Mirror 04:
List of pools:   pool1
List of disks:   disk7 disk8

Secondary Tier
==============
Size:            20.00G
Use%:            0%
Layout:          mirrored
Mirrors:         2
Columns:         -
Stripe Unit:     0.00 K
FastResync:      Disabled

1. Mirror 01:
List of pools:   pool1
List of disks:   disk9 disk10

2. Mirror 02:
List of pools:   pool1
List of disks:   disk11 disk12

FS Type:           Normal

Defrag Status: Not Running
Fullfsck Status: Not Running
Resync Status:
    Tier 1, Mirror 03: 7.67%     Start_time: Jun/05/2011/15:10:26   Work_time: 0:1:14
    Remaining_time: 14:50
    Tier 1, Mirror 04: 1.08%     Start_time: Jun/05/2011/13:08:07   Work_time: 0:1:21
    Remaining_time: 2:03:33
    Tier 2, Mirror 02: 11.27%    Start_time: Jun/05/2011/14:03:14   Work_time: 0:0:18
    Remaining_time: 02:21
```

```
Rollsync Status:
    Rollsync  mir3_roll2, Tier 1: 1.28%  Start_time: Jun/05/2011/14:51:40   Work_
↪time: 0:0:12     Remaining_time: 15:23
Relayout Status: Not Running
```

Make a file system online.

```
Storage> fs online fs1
100% [#] Online filesystem
```

Check and repair a file system while it is in offline state.

```
Storage> fs fsck fs1
ACCESS fs ERROR V-288-693 fs1 must be offline to perform fsck.
```

Grow file system's primary tier size to 1 G.

```
Storage> fs growto primary fs1 1g
```

Grow file system's primary tier size by 50 M.

```
Storage> fs growby primary fs1 50M
```

Shrink file system's primary tier by specified size.

```
Storage> fs shrinkby primary fs1 10m
```

Shrink file system's primary tier to just 50 M.

```
Storage> fs shrinkto primary fs1 50M
```

View the list of file systems.

```
Storage> fs list
FS   STATUS       SIZE    LAYOUT   MIRRORS   COLUMNS   USE%    NFS      CIFS      FTP␣
↪     SECONDARY
===  ======       ====    ======   =======   =======   ====  =======  =======  ␣
↪=======    ==========
fs1  online    100.00M    simple      -         -       3%     no        no       no ␣
↪        no
fs2  online    100.00M    simple      -         -       3%     no        no       no ␣
↪        no
fs3  online    700.00M    simple      -         -       4%     no        no       no ␣
↪        no
fs4  online     69.00M    simple      -         -       4%     no        no       no ␣
↪        no
fs5  online      1.94G    simple      -         -       1%     no        no       no ␣
↪        yes
fs6  online      1.94G    simple      -         -       1%     no        no       no ␣
↪        yes,metaOk
```

Display detailed information for a specific file system.

```
Storage> fs list fs6
General Info:
===============
Block Size:      1024 Bytes
Version:         Version 11
```

```
node_01:         online
node_02:         offline

Primary Tier
============
Size:            500.00M
Use%:            0%
Layout:          mirrored
Mirrors:         2
Columns:         -
Stripe Unit:     0.00 K
FastResync:      Enabled

1. Mirror 01:
List of pools:   p03
List of disks:   ams_wms0_12

2. Mirror 02:
List of pools:   p03
List of disks:   ams_wms0_13

FS Type:           Normal

Defrag Status:   Not Running
Fullfsck Status: Not Running
Resync   Status: Not Running
RollSync Status: Not Running
Relayout Status: Not Running
```

Disable FastResync for a file system.

```
Storage> fs unsetfastresync fs6
```

Try disabling FastResync for a file system where it is already in disabled state.

```
Storage> fs unsetfastresync fs6
ACCESS fs ERROR V-288-655 Fastresync is not enabled for fs6.
```

Enable FastResync for a file system.

```
Storage> fs setfastresync fs6
```

Try enabling FastResync for a file system where it is already in enabled state.

```
Storage> fs setfastresync fs6
ACCESS fs ERROR V-288-651 File system fs6 is already fastresync enabled.
```

Defragment a file system.

```
Storage>fs defrag fs0 1H10M
The file system fs0 is offline, please online fs0 before defragmentation

Storage>fs defrag fs1 1H10M
it will take some time to do the defragmentation
do you want to continue? yes|no
n
Defragmentation cancelled for fs1
Storage>fs defrag fs1 1H10M
```

```
it will take some time to do the defragmentation
do you want to continue? yes|no
y
Storage>
```

You can run other command, and you can check the defragment status by

```
Storage>fs list fs1
General Info:
===============
Block Size:      1024 Bytes
Version:         Version 11
node_01:        offline
node_02:        offline

Primary Tier
============
Size:            1.00G
Use%:            -
Layout:          simple
Mirrors:         -
Columns:         -
Stripe Unit:     0.00 K
FastResync:      Disabled

Mirror 1:
List of pools:   pool
List of disks:   disk_4 disk_5

FS Type:           Normal

Defrag Status:   Done successfully
Fullfsck Status: Not Running
```

Destroy a file system.

```
Storage> fs destroy fs6
100% [#] Destroy filesystem

FS
============
fs1
```

Set alert on the file system.

```
Storage> fs alert set numinodes 2M fs2
ACCESS fs SUCCESS V-288-663 Alert of type [ numinodes ] set to 2M on the file system␣
↪fs2
```

Unset the alert on the file system.

```
Storage> fs alert unset numinodes fs2
ACCESS fs SUCCESS V-288-663 Alert of type [ numinodes ] set to DEFAULT value on the␣
↪file system fs2
```

Set alert on a snapshot of the file system.

```
Storage> fs alert set numinodes 2M fs2 snap1
ACCESS fs SUCCESS V-288-663 Alert of type [ numinodes ] set to 2M on the file system␣
↪fs2 snapshot snap1
```

Unset the alert on a snapshot of the file system.

```
Storage> fs alert unset numinodes fs2 snap1
ACCESS fs SUCCESS V-288-663 Alert of type [ numinodes ] set to DEFAULT value on the␣
↪file system fs2 snapshot snap1
```

Set the default alert.

```
test_01.Storage> fs alert set numinodes 2M
ACCESS fs SUCCESS V-288-663 Default Alert on the file systems of type [ numinodes ]␣
↪set to 2M
```

Unset the default alert.

```
test_01.Storage> fs alert unset numinodes
ACCESS fs SUCCESS V-288-663 Alert of type [ numinodes ] set to default value 0
```

Show the alert status.

```
Storage> fs alert show

File System          Alert Type      Value       Current Usage
==================== ==========      =========       =============
fs0                  numspace        80% (D)         2%
fs0                  numinodes        6500            1000
fs0                  fullspace       98% (D)         2%
fs1                  numspace        80% (D)         2%
fs1                  numinodes        8000            10000
fs1                  fullspace       98% (D)         2%
fs4                  numspace        80(D)%          3%
fs4                  numinodes        2000000 (D)     4
fs4                  fullspace       98% (D)         3%
```

Upgrade the file system layout to the latest layout.

```
Storage> fs upgrade fs1
Upgrading the file system, are you sure that you want to upgrade the file
system, the operation is irreversible, please enter yes/no
yes

ACCESS fs INFO V-288-2578 Proceeding with fs upgrade operation ...
    ACCESS fs SUCCESS V-288-2474 file system upgraded to version 13.


Storage> fs upgrade fs1
ACCESS fs ERROR V-288-2471 File system fs1 is already at upgraded version 13.

Storage> fs upgrade fs2
ACCESS fs ERROR V-288-2482 fs2 must be online to perform upgrade operation.



Storage> fs list fs list fs1
    General Info:
```

```
    ===============
    Block Size:        8192 Bytes
    Version:           Version 13
    Volume Encrypted:  No
    Max IOPS:  0
    isaC_01:        online
    isaC_02:        online

    Primary Tier
    ============
    Size:            1.00G
    Use%:            5%
    Layout:          simple
    Mirrors:         -
    Columns:         -
    Stripe Unit:     0.00 K
    Meta Data:       metaOk
    FastResync:      Disabled

    1. Mirror 01:
    List of pools:   tpool
    List of disks:   emc0_0172 emc0_0173 emc0_0174

    FS Type:           Normal

    Defrag Status: Not Running
    Fullfsck Status: Not Running
    Resync   Status: Not Running
    Rollsync Status: Not Running
    Relayout Status: Not Running

    WORM Enabled: No
```

Create a schedule by name defrag_sched_1 that runs compression jobs at 11:00 pm every Saturday. This defrag job should run only for 8 hours.

```
Storage>fs defrag schedule create defrag_sched_1 8 15 23 * * 6
Storage>Defrag Schedule creation succeeded
```

Show the defarg schedule details of the schedule by name defrag_sched_1.

```
Storage> fs defrag schedule show defrag_sched_1
Name            Node       Duration  Minute Hour   Day      Month    WeekDay
====            ====       ========  ====== ====   ===      =====    ========
defrag_sched_1   any       8 hours   15     23     *        *        6
```

Start defrag job for file system tpcc_data1 with schedule by name defrag_sched_1.

```
Storage> fs defrag schedule start tpcc_data1 defrag_sched_1
Storage>starting defrag_sched_1 for fsname tpcc_data1
```

List the scheduled compression job status for file system tpcc_data1.

```
Storage> fs defrag schedule list tpcc_data1
Schedule Information for tpcc_data1
============================================
Name            Node       Duration   Minute Hour   Day      Month    WeekDay
```

```
====            ====       ========   ====== ====   ===      =====    ========
defrag_sched_1  nasvm67_0   1 hour(s)  15     *      *        *        *
```

Create a file system for a virtual machine workload with 4 striped columns.

```
Storage> fs create pretuned vmdk_fs 100g pool1  workload=virtualmachine␣
↪layout=striped 4
```

Create a file system for a media server workload with 8 striped columns.

```
Storage> fs create pretuned media_fs 100g pool2  workload=mediaserver layout=striped-
↪mirror 8
```

List the file system properties of vmdk_fs.

```
storage> fs list vmdk_fs
General Info:
===============
Block Size:       8192 Bytes
Version:          Version 11
Workload:         virtualmachine datastore
Extent Size:      1m
ISAGA_01:        online
ISAGA_02:        online

Primary Tier
============
Size:            100G
Use%:            5%
Layout:          striped
Mirrors:         -
Columns:         4
Stripe Unit:     512 K
Metadata:        metaOk
FastResync:      Disabled
```

Create a simple file system on encrypted volume

```
Storage> fs create simple fs4 1g pool1 blksize=2048 pdir_enable=no encrypt=on
100% [#] Creating simple filesystem
ACCESS fs SUCCESS V-288-0 Created simple file system fs4
```

Display detailed information for a specific file system

```
Storage> fs list fs4
General Info:
===============
Block Size:        2048 Bytes
Version:           Version 11
Volume Encrypted:  Yes
ISA_01:        online
ISA_02:        online

Primary Tier
============
Size:            1.00G
Use%:            5%
Layout:          simple
```

```
Mirrors:         -
Columns:         -
Stripe Unit:     0.00 K
Meta Data:       metaOk
FastResync:      Disabled

1. Mirror 01:
List of pools:   pool1
List of disks:   isa_01_intel_nvme2_0

FS Type:           Normal

Defrag Status: Not Running
Fullfsck Status: Not Running
Resync   Status: Not Running
Rollsync Status: Not Running
Relayout Status: Not Running

WORM Enabled: No
```

Create a simple WORM enabled file system

```
Storage> fs create simple fs1 1g pool1 blksize=2048 pdir_enable=no encrypt=off␣
↪worm=yes
100% [#] Creating simple filesystem
ACCESS fs SUCCESS V-288-0 Created simple file system fs1
```

Display detailed information for a specific file system

```
Storage> fs list fs1
General Info:
===============
Block Size:        2048 Bytes
Version:           Version 13
Volume Encrypted:  No
Max IOPS:  0
ISA_01:        online
ISA_02:        online

Primary Tier
============
Size:            1.00G
Use%:            5%
Layout:          simple
Mirrors:         -
Columns:         -
Stripe Unit:     0.00 K
Meta Data:       metaOk
FastResync:      Disabled

1. Mirror 01:
List of pools:   pool1
List of disks:   isa_01_intel_nvme2_0

FS Type:           Normal

Defrag Status: Not Running
Fullfsck Status: Not Running
```

```
Resync   Status: Not Running
Rollsync Status: Not Running
Relayout Status: Not Running

WORM Enabled: Yes
```

Create a data movement policy policy1 for file system fs1 to move the files with file name extensions of .txt and .pdf from the primary tier (disk tier) to tier1 (cloud tier), which did not get accessed or modified for the last 2 days.

```
Storage> fs policy add operation=move policy1 fs1 primary tier1 *.txt,*.pdf atime >2d␣
↪mtime >2d
ACCESS policy SUCCESS V-288-0 Policy policy1 for fs fs1 added successfully.
```

Retrieve data from Amazon Glacier. Create a policy pol1 to move all the files with file name extension of .txt from Amazon Glacier to the primary tier using the Bulk retrieval option. Files are copied to on-premises and then deleted from Amazon Glacier. The time when the files are available on-premises depends on the type of retrieval option selected.

```
Storage> fs policy add operation=move pol1 gfs2 gtier primary retrieval_option=Bulk␣
↪\*.txt.
```

Create a data deletion policy policy2 for file system fs1 to move the files with file name extensions of .txt and .pdf from tier1 (cloud tier), which did not get accessed or modified for the last 2 days.

```
Storage> fs policy add operation=delete policy2 fs1 tier1 \*.txt,\*.pdf atime >2d␣
↪mtime >2d
ACCESS policy SUCCESS V-288-0 Policy policy2 for fs fs1 added successfully.
```

Modify data movement policy policy1 for file system fs1 to move the files with file name extension of .doc, which did not get accessed or modified for the last 3 days.

```
Storage> fs policy modify policy1 \*.doc atime >3d mtime >3d
ACCESS policy SUCCESS V-288-0 Policy policy1 modified successfully.
```

List all policies.

```
Storage> fs policy list
Name     FS name   Action   Source Tier      Destination Tier   Retrieval Option  ␣
↪Pattern       Atime   Mtime   State
=======  ========  =======  ===============  =================  =================  ␣
↪============  ======  ======  ===========
policy2  fs1       delete   tier1            -                  Standard           \*.
↪txt, \*.pdf  >2d     >2d     not running
policy1  fs1       move     primary          tier1              Standard           \*.
↪doc         >3d     >3d     running
```

List all policies set for file system fs1.

```
Storage> fs policy list fs1
Name     FS name   Action   Source Tier      Destination Tier   Retrieval Option  ␣
↪Pattern       Atime   Mtime   State
=======  ========  =======  ===============  =================  =================  ␣
↪============  ======  ======  ===========
policy2  fs1       delete   tier1            -                  Standard           \*.
↪txt, \*.pdf  >2d     >2d     running
policy1  fs1       move     primary          tier1              Standard           \*.
↪doc         >3d     >3d     not running
```

Delete policy policy1 set for file system fs1.

```
Storage> fs policy delete policy1 fs1
ACCESS policy SUCCESS V-288-0 Policy policy1 for fs fs1 deleted successfully.
```

Rename policy2 to policy3.

```
Storage> fs policy rename policy2 policy3
ACCESS policy SUCCESS V-288-0 Policy policy2 renamed to policy3.
```

Show the status of policy run for the policy *Policy1*.

```
storage> fs policy status Policy1
Policy Name:                    Policy1
==================================================
Policy Run Type:                normal
Policy Run Status:              running
Total Data (Files):             93.1 GB (100000)
Moved/Deleted Data (Files):     47.7 MB (879)
Last File Visited:              file100.txt
```

Abort the currently running policy *Policy1*.

```
storage> fs policy abort Policy1
ACCESS policy INFO V-288-0 Policy Policy1 aborted successfully.
```

Start a dry run of the policy *Policy1*.

```
storage> fs policy dryrun Policy1
ACCESS policy INFO V-288-0 Policy Policy1 dryrun started in background, please check
↪'fs policy status' for progress.
```

Pause the currently running policy *Policy1*

```
storage> fs policy pause Policy1
ACCESS policy INFO V-288-0 Policy Policy1 paused successfully.
```

Run the currently paused policy *Policy1*.

```
storage> fs policy run Policy1
Policy Policy1 is not running currently, as it was killed/paused. Would you like to␣
↪start new run (y/n): y
ACCESS policy INFO V-288-0 Policy Policy1 run started in background, please check 'fs␣
↪policy status' for progress.
```

Resume the currently paused policy *Policy1*.

```
storage> fs policy resume Policy1
ACCESS policy INFO V-288-0 Policy Policy1 resume started in background, please check
↪'fs policy status' for progress.
```

Create the schedule for the file system *lfs1*. The schedule runs every 10 minutes.

```
storage> fs policy schedule create lfs1 */10 * * * *
ACCESS policy_schedule SUCCESS V-288-0 Schedule create for file system lfs1 done␣
↪successfully.
```

Modify the schedule of the file system *lfs1*. The schedule now runs every 20 minutes.

```
storage> fs policy schedule modify lfs1 */20 * * * *
ACCESS policy_schedule SUCCESS V-288-0 Schedule modify for file system lfs1 done␣
↪successfully.
```

List the schedule of the file system *lfs1*.

```
storage> fs policy schedule list lfs1
File System Name   Minute   Hour   Day of Month   Month   Day of Week
=================  =======  =====  =============  ======  ============
lfs1               20       *      *              *       *
```

Remove the schedule of the file system *lfs1*.

```
storage> fs policy schedule remove lfs1
ACCESS policy_schedule SUCCESS V-288-0 Schedule for file system lfs1 removed␣
↪successfully.
```

Set the retention on the file */vx/myfs/file1 5y*.

```
Storage> storage fs retention set /vx/myfs/file1 5y
ACCESS Retention SUCCESS V-288-0 Successfully set retention on /vx/myfs/file1
```

Set the retention on the file */vx/myfs/file1 05-20-2020*.

```
Storage> storage fs retention set /vx/myfs/file1 05-20-2020
ACCESS Retention SUCCESS V-288-0 Successfully set retention on /vx/myfs/file1
```

Set the retention on the directory */vx/myfs/dir1 2y*.

```
Storage> storage fs retention set /vx/myfs/dir1 2y
ACCESS Retention SUCCESS V-288-0 Applying retention on all current files in directory,
↪ this will take some time. Check report showevent for status.
```

Set the retention on the directory */vx/myfs/dir1 05-20-2020*.

```
Storage> storage fs retention set /vx/myfs/dir1 05-20-2020
ACCESS Retention SUCCESS V-288-0 Applying retention on all current files in directory,
↪ this will take some time. Check report showevent for status.
```

Show the retention on the file */vx/myfs/file1*

```
Storage> storage fs retention show /vx/myfs/file1
ACCESS Retention SUCCESS V-288-0 Retention value on file /vx/myfs/file1 is 06-08-2022␣
↪23:12:51
```

Show the retention on the directory */vx/myfs/dir1*

```
Storage> storage fs retention show /vx/myfs/dir1
ACCESS Retention ERROR V-288-0 Specified path is directory path, please give file path
```

Clear the retention on the file */vx/myfs/file1*

```
Storage> storage fs retention clear /vx/myfs/file1
ACCESS Retention SUCCESS V-288-0 Successfully cleared retention on /vx/myfs/file1
```

Clear the retention on the directory */vx/myfs/dir1*

```
Storage> storage fs retention clear /vx/myfs/dir1
ACCESS Retention SUCCESS V-288-0 Clearing retention of all current file in directory,␣
↪this will take some time. Check report showevent for status.
```

Enable worm support for specified file system.

```
Storage> fs worm set fs1
ACCESS fs SUCCESS V-288-0 Enabled WORM for fs1 file system.
```

Disable worm support for specified file system.

```
Storage> fs worm clear fs1
ACCESS fs SUCCESS V-288-0 Disabled WORM for fs1 file system.
```

17.7.6 SEE ALSO

disk(1), hba(1), fencing(1), pool(1), scanbus(1), snapshot(1), tier(1), iscsi(1), quota(1) maxiops(1)

17.8 hba

17.8.1 SYNOPSIS

`hba` [*host_name*]

17.8.2 DESCRIPTION

The storage `hba` command displays WWN, state, and some other information for the specified node. WWN: World Wide Name State: online/offline Speed: Per Second Transmitted_FC_Frames: A value equal to the number of total transmitted Serial Attached SCSI frames across all the protocols. Received_FC_frames: A value equal to the number of total received Serial Attached SCSI frames across all the protocols. Link_Failure_Count: A value equal to the value of the LINK FAILURE COUNT field of the Link Error Status

17.8.3 EXAMPLES

Display WWN information for a particular node.

```
Storage> hba democluster_01
HBA_Node_Name            WWN                      State    Speed Support_Classes␣
↪Transmitted_FC_Frames Received_FC_frames  Link_Failure_Count
-------------            ---                      -----    ----- --------------- -----
↪---------------- ------------------  ------------------
20:00:00:1b:32:89:15:5f  21:00:00:1b:32:89:15:5f  offline  4_Gbit  Class_3        ␣
↪445606                  1815671                1
20:01:00:1b:32:a9:15:5f  21:01:00:1b:32:a9:15:5f  offline  unknown Class_3         0 ␣
↪                    0                      0
B.Storage>
democluster_01      21:00:00:1b:32:1e:5c:ba, 21:01:00:1b:32:3e:5c:ba
```

Display WWN information for all the running nodes in the cluster.

```
Storage> hba
Node            Host Initiator HBA WWNs
----            -----------------------
democluster_01      21:00:00:1b:32:89:15:5f, 21:01:00:1b:32:a9:15:5f
democluster_02      21:00:00:1b:32:89:71:52, 21:01:00:1b:32:a9:71:52
```

17.8.4 SEE ALSO

disk(1), fencing(1), fs(1), pool(1), scanbus(1), snapshot(1), tier(1), iscsi(1), quota(1)

17.9 iscsi

17.9.1 SYNOPSIS

`iscsi status`

`iscsi start`

`iscsi stop`

`iscsi device add` *device*

`iscsi device del` *device*

`iscsi device list`

`iscsi discovery add` *discovery-address*

`iscsi discovery del` *discovery-address*

`iscsi discovery rediscover` *discovery-address*

`iscsi discovery rediscover_new` *discovery-address*

`iscsi discovery list`

`iscsi target add` *target-name portal-address*

`iscsi target del` *target-name* { *discovery-address* | *portal-address* }

`iscsi target login` *target-name* { *discovery-address* | *portal-address* }

`iscsi target logout` *target-name* { *discovery-address* | *portal-address* }

`iscsi target rescan` *target-name*

`iscsi target attr showdefault`

`iscsi target attr showall`

`iscsi target attr show` *target-name*

`iscsi target attr setdefault` *attribute value*

`iscsi target attr setall` *attribute value*

`iscsi target attr set` *target-name attribute value*

`iscsi target list`

`iscsi target listdetail` *target-name*

`iscsi initiatorname setprefix` *initiatorname-prefix*

`iscsi initiatorname list`

17.9.2 TARGET ATTRIBUTES

login_timeout The amount of time the iSCSI initiator service should wait for a login to complete. The value of this attribute is in seconds. Valid values range from 10 to 600.

logout_timeout The amount of time the iSCSI initiator service should wait for logout to complete. The value of this attribute is in seconds. Valid values range from 10 to 600.

initial_login_retry_max The maximum number of times the iSCSI initiator should try a login to the target during the first login. This only affects the initial login. Valid values range from 1 to 16. During each login attempt wait for *login_timeout* seconds, for the login to succeed.

noop_timeout The amount of time the iSCSI initiator service should wait for response to a Nop-out request sent to the target, before failing the connection. Failing the connection would result in I/O to be failed and retried on any other available path. The value of this attribute is in seconds. Valid values range from 5 to 600.

noop_interval The time to wait between subsequent sending of Nop-out requests. The value of this attribute is in seconds. Valid values range from 5 to 600.

replacement_timeout The amount of time to wait for session re-establishment before failing SCSI commands. The value of this attribute is in seconds. Valid values range from 10 to 86400.

cmds_max The maximum number of SCSI commands the session queues. A session is defined as a connection between the initiator and the target portal for accessing a given target. *cmds_max* defines the commands per target, which could be multiple LUNs. Valid values range from 2 to 2048 and should be a power of 2.

queue_depth The maximum number of SCSI commands queued per LUN, belonging to a target. The value for *queue_depth* cannot be greater than *cmds_max*. Valid values range from 1 to 128.

fast_abort Defines whether the initiator should respond to R2Ts (Request to Transfer) after sending a task management function like an ABORT_TASK or LOGICAL UNIT RESET. A value of 'Yes' causes the initiator to stop responding to R2Ts after an ABORT_TASK request is received. For Equallogic arrays, the recommended value is 'No'. Valid values are Yes/No.

17.9.3 OPTIONS

device Device on which the operation takes place.

target-name Name of the iSCSI target at which SCSI LUNs are available. *target-name* should conform to the naming rules defined in RFC3721.

discovery-address Discovery address is the target address at which the initiator can request a list of targets using SendTargets text request as specified in the iSCSI protocol of RFC3720. If no port is specified with the discovery address, the default port 3260 is used. IPv4 and IPv6 addresses are supported.

Examples

```
192.168.0.4
192.168.0.4:3260
2001:c90::211:9ff:feb8:a9e9
[2001:c90::211:9ff:feb8:a9e9]:3260
```

initiatorname-prefix initiatorname-prefix is a name that conforms to the naming rules for initiator and target names as specified in RFC3721. Initiator names for nodes in the cluster are generated by appending the node number to this prefix.

portal-address Portal address is the location at which the target is accessible. IPv4 and IPv6 addresses are supported

Examples:

```
192.168.0.4
192.168.0.4,1
192.168.0.4:3260
192.168.0.4:3260,1
```

```
2001:c90::211:9ff:feb8:a9e9
2001:c90::211:9ff:feb8:a9e9,1
[2001:c90::211:9ff:feb8:a9e9]:3260
[2001:c90::211:9ff:feb8:a9e9]:3260,10
```

iscsi status Show the status of the iSCSI initiator service.

iscsi start Start the iSCSI initiator service.

iscsi stop Stops the iSCSI initiator service.

iscsi device add ***device*** Add a device for use with the iSCSI initiator. iSCSI initiator connections use this device to connect to the target. If there are any existing targets, then the iSCSI initiator initiates a connection to all the targets via *device*.

iscsi device del ***device*** Delete a *device* from the iSCSI initiator configuration. Any existing connections via *device* to targets are terminated. If *device* is the last device in the iSCSI initiator configuration and there are existing targets, then the device cannot be deleted from the configuration.

iscsi device list List the devices used by the iSCSI initiator.

iscsi discovery add ***discovery-address*** Add a discovery address to the iSCSI initiator configuration. If no TCP port is specified with the *discovery-address*, then the default port 3260 is used. Any targets discovered at *discovery-address* are automatically logged on to.

iscsi discovery del ***discovery-address*** Delete a discovery address from an iSCSI initiator configuration. Any targets discovered using *discovery-address* are also deleted from the configuration.

iscsi discovery rediscover ***discovery-address*** Perform a discovery of changes in targets or LUNs at *discovery-address*. Any LUNs or targets that have been removed at *discovery-address* are automatically removed from the configuration. New LUNs or targets discovered at *discovery-address* are automatically added and logged on to.

iscsi discovery rediscover_new ***discovery-address*** Perform a discovery of changes in targets or LUNs at *discovery-address*. And new LUNs or targets discovered at *discovery-address* are automatically added and logged on to. Does not discover any targets that have been deleted at fIdiscovery- addressfP.

iscsi discovery list List the discovery addresses present in the iSCSI initiator configuration.

iscsi target add ***target-name portal-address*** Add a static target-portal combination to the iSCSI initiator configuration. The *portal-address* cannot be the same as any *discovery-address* present in the iSCSI initiator configuration. Connections to *portal-address* are made for *target-name*, but no discovery is done for any other targets available at *portal-address*. If no portal tag is specified with *portal-address*, the default portal tag of 1 is used.

iscsi target del ***target-name* { *discovery-address* | *portal-address* }** Delete a target *target-name* from the iSCSI initiator configuration. Any existing connections to *target-name* are terminated. *discovery-address* or *portal-address* is the address through which the target became visible to the initiator. A target that was discovered at a *discovery-address* once deleted from the iSCSI initiator configuration is again visible to the iSCSI initiator if re-discovery is done either through `iscsi discovery rediscover` or `scanbus` commands.

iscsi target login ***target-name* { *discovery-address* | *portal-address* }** Log on to a target *target-name* from the iSCSI initiator. Connections to *target-name* are made from all devices present in the iSCSI initiator configuration. *discovery-address* or *portal-address* is the address through which the target became visible to the initiator.

iscsi target logout ***target-name* { *discovery-address* | *portal-address* }** Log out from connections to *target-name* from the iSCSI initiator. *discovery-address* or *portal-address* is the address

through which the target became visible to the initiator. A target once logged out by the iSCSI initiator is not logged on to until `iscsi target login` is requested.

iscsi target rescan ***target-name*** Rescan target *target-name* for new LUNs.

iscsi target attr showdefault Show the default value for target attributes.

iscsi target attr showall Show the value of target attributes for all the known targets.

iscsi target attr show ***target-name*** Show the value of target attributes for target *target-name*

iscsi target attr setdefault ***attribute value*** Set the default value of *attribute* to *value*. Default value is inherited by any new targets that get added.

iscsi target attr setall ***attribute value*** Set the value of *attribute* to *value* for all the known targets. This does not change the default value as shown in `iscsi target attr showdefault`. Changes to the values are effective after re-login.

iscsi target attr set ***target-name attribute value*** Set the value of *attribute* to *value* for *target-name*. Changes to the values are effective after re-login.

iscsi target list List the targets visible to the iSCSI initiator. A target can be in any one of ONLINE/OFFLINE/RETRY states.

iscsi target listdetail ***target-name*** List detailed information about target *target-name*.

iscsi initiatorname setprefix ***initiatorname-prefix*** Set the prefix used to generate initiator names. Initiator names are generated as *initiatorname-prefix* followed by the node number of the node.

iscsi initiatorname list List the initiator names for all the nodes in the cluster.

17.9.4 EXAMPLES

Display the status of the iSCSI initiator service.

```
Storage> iscsi status
iSCSI Initiator Status on test_1 : ONLINE
iSCSI Initiator Status on test_2 : ONLINE
```

Stop the iSCSI initiator service.

```
Storage> iscsi stop
Success.
Storage> iscsi status
iSCSI Initiator Status on test_1 : OFFLINE
iSCSI Initiator Status on test_2 : OFFLINE
```

Start the iSCSI initiator service.

```
Storage> iscsi start
Success.
Storage> iscsi status

iSCSI Initiator Status on test_1 : ONLINE
iSCSI Initiator Status on test_2 : ONLINE
```

Set the initiator name prefix used to generate the iSCSI initiator names.

```
Storage> iscsi initiatorname setprefix iqn.2009-05.com.test:test
Success.
Storage>
```

List the initiator names of all the nodes.

```
Storage> iscsi initiatorname list
Node              Initiator Name
----              --------------
test_1             iqn.2009-05.com.test:test.1
test_2             iqn.2009-05.com.test:test.2
```

List the devices used by the iSCSI initiator.

```
Storage> iscsi device list
Device
------
pubeth0
pubeth1
```

Delete a device from the iSCSI initiator configuration.

```
Storage> iscsi device del pubeth1
100% [#] Updating disk list
Storage> iscsi device list

Device
------
pubeth0
```

Add a device to the iSCSI initiator configuration.

```
Storage> iscsi device add pubeth1
100% [#] Updating disk list
Storage> iscsi device list

Device
------
pubeth0
pubeth1
```

List the discovery addresses known to the iSCSI initiator.

```
Storage> iscsi discovery list

Discovery Address
-----------------
192.168.2.14:3260
192.168.2.15:3260
```

Delete a discovery address from the iSCSI initiator configuration. Any targets discovered using the discovery addresses are automatically deleted.

```
Storage> iscsi discovery del 192.168.2.15:3260
100% [#] Updating disk list
Storage> iscsi discovery list

Discovery Address
```

```
-----------------
192.168.2.14:3260
```

Add a discovery address to the iSCSI initiator configuration. Any discovered targets are automatically logged in to.

```
Storage> iscsi discovery add 192.168.2.15:3260
17% [-] Adding discovery 192.168.2.15:3260

Discovery CHAP credentials for test_1:
Outgoing CHAP Username : root
Outgoing CHAP Password : ********
Incoming CHAP Username :
Authentication succeeded.

Discovered Targets
-----------------
iqn.2001-04.com.example:storage.disk2.sys3.xyz
iqn.2001-04.com.example:storage.disk3.sys3.xyz
iqn.2001-04.com.example:storage.disk4.sys3.xyz
iqn.2001-04.com.example:storage.disk5.sys3.xyz

20% [/] Updating target configuration

Logging into target iqn.2001-04.com.example:storage.disk2.sys3.xyz
Logging into target iqn.2001-04.com.example:storage.disk3.sys3.xyz
Logging into target iqn.2001-04.com.example:storage.disk4.sys3.xyz
Logging into target iqn.2001-04.com.example:storage.disk5.sys3.xyz

100% [#] Updating disk list
Storage> iscsi discovery list

Discovery Address
-----------------
192.168.2.14:3260
192.168.2.15:3260
```

Discover changes in LUNs or targets at a discovery address already added to the iSCSI initiator configuration. Any discovered targets are automatically logged in to.

```
Storage> iscsi discovery rediscover 192.168.2.15:3260
12% [/] Checking for deleted targets

Deleted targets
-----------------
iqn.2001-04.com.example:storage.disk5.sys3.xyz

Deleting iqn.2001-04.com.example:storage.disk5.sys3.xyz...done

14% [|] Checking for new targets

New targets
-----------------
iqn.2001-04.com.example:storage.disk6.sys3.new.xyz

100% [#] Updating disk list
```

Discover new LUNs or targets at a discovery address already added to the iSCSI initiator configuration. Any discovered targets are automatically logged in to.

```
Storage> iscsi discovery rediscover_new 192.168.2.15:3260
14% [|] Checking for new targets

New targets
-----------------
iqn.2001-04.com.example:storage.disk7.sys3.new.xyz

100% [#] Updating disk list
```

Add an IPv6 discovery address.

```
Storage> iscsi discovery add fec0::20c:29ff:fe8e:2200
17% [-] Adding discovery fec0::20c:29ff:fe8e:2200

Discovered Targets
-----------------
iqn.2001-04.example.storage:disk.01
iqn.2002-04.example.storage:disk.02
iqn.2003-04.example.storage:disk.03

20% [/] Updating target configuration

Logging into target iqn.2001-04.example.storage:disk.01
Logging into target iqn.2002-04.example.storage:disk.02
Logging into target iqn.2003-04.example.storage:disk.03

100% [#] Updating disk list
Storage> iscsi discovery list

Discovery Address
-----------------
[fec0::20c:29ff:fe8e:2200]:3260

Storage>
```

List the targets that are configured on the iSCSI initiator.

```
Storage> iscsi target list

Target                                              Discovery Address     State      ␣
↪Disk
------                                              -----------------     -----      -
↪---
iqn.2001-04.com.example:storage.disk2.sys3.xyz      192.168.2.14:3260     ONLINE     ␣
↪disk_0
iqn.2001-04.com.example:storage.disk4.sys3.xyz      192.168.2.14:3260     ONLINE     ␣
↪disk_2
iqn.2001-04.com.example:storage.disk5.sys3.xyz      192.168.2.14:3260     ONLINE     ␣
↪disk_3
iqn.2001-04.com.example:storage.disk3.sys3.xyz      192.168.2.14:3260     ONLINE     ␣
↪disk_1
iqn.2001-04.com.example2:storage.disk2.sys3.xyz     192.168.2.15:3260     ONLINE     ␣
↪disk_4
iqn.2001-04.com.example2:storage.disk3.sys3.xyz     192.168.2.15:3260     ONLINE     ␣
↪disk_5
iqn.2001-04.com.example2:storage.disk4.sys3.xyz     192.168.2.15:3260     ONLINE     ␣
↪disk_6
iqn.2001-04.com.example2:storage.disk5.sys3.xyz     192.168.2.15:3260     ONLINE     ␣
↪disk_7
```

List details of connections to a target.

```
Storage> iscsi target listdetail
iqn.2001-04.com.example:storage.disk2.sys3.xyz

Discovery Address : 192.168.2.14:3260

Connections
===========

Portal Address           test_1   test_2
--------------           -----   -----
192.168.2.14:3260,1      2       2
```

Add a target using the portal address without using automatic discovery.

```
Storage> iscsi target add iqn.2001-04.com.example:storage.disk2.sys1.xyz 192.168.2.
↪14:3260
```

Logging into target iqn.2001-04.com.example:storage.disk2.sys1.xyz.

```
Storage> iscsi target listdetail iqn.2001-04.com.example:storage.disk2.sys1.xyz

Connections
===========

Portal Address        test55_01   test55_02
--------------        --------   --------
192.168.2.14:3260,1   1          1
```

Add an IPv6 target using the portal address without using automatic discovery.

```
Storage> iscsi target add iqn.2001-04.ngsfdellpe-04.storage:disk.01␣
↪[fec0::20c:29ff:fe8e:2200]:3260,1
100% [#] Updating disk list
Storage> iscsi target listdetail iqn.2001-04.ngsfdellpe-04.storage:disk.01

Connections
===========

Portal Address                       test55_01    test55_02
--------------                       --------    --------
[fec0::20c:29ff:fe8e:2200]:3260,1    2           2

Storage>
```

Logout connections to a target.

```
Storage> iscsi target logout iqn.2001-04.com.example:storage.disk2.sys3.xyz
100% [#] Updating disk list
Storage> iscsi target listdetail iqn.2001-04.com.example:storage.disk2.sys3.xyz

Discovery Address : 192.168.2.14:3260

Connections
===========
```

```
Portal Address           test_1   test_2
--------------           -----   -----
192.168.2.14:3260,1      0       0
```

Log on to a target.

```
Storage> iscsi target login iqn.2001-04.com.example:storage.disk2.sys3.xyz
100% [#] Updating disk list
Storage> iscsi target listdetail iqn.2001-04.com.example:storage.disk2.sys3.xyz

Discovery Address : 192.168.2.14:3260

Connections
===========

Portal Address           test_1   test_2
--------------           -----   -----
192.168.2.14:3260,1      2       2
```

Delete a target from the iSCSI initiator configuration.

```
Storage> iscsi target del iqn.2001-04.com.example:storage.disk2.sys3.xyz
100% [#] Updating disk list
```

Rescan target for new LUNs.

```
Storage> iscsi target rescan iqn.2001-04.com.example:storage.disk2.sys3.xyz
100% [#] Updating disk list
Storage> iscsi target list

Target                                              Discovery Address     State     ␣
↪Disk
------                                              -----------------     -----      -
↪---
iqn.2001-04.com.example:storage.disk2.sys3.xyz      192.168.2.14:3260     ONLINE    ␣
↪disk_0 disk_8 disk_9
iqn.2001-04.com.example:storage.disk4.sys3.xyz      192.168.2.14:3260     ONLINE    ␣
↪disk_2
iqn.2001-04.com.example:storage.disk5.sys3.xyz      192.168.2.14:3260     ONLINE    ␣
↪disk_3
iqn.2001-04.com.example:storage.disk3.sys3.xyz      192.168.2.14:3260     ONLINE    ␣
↪disk_1
iqn.2001-04.com.example2:storage.disk2.sys3.xyz     192.168.2.15:3260     ONLINE    ␣
↪disk_4
iqn.2001-04.com.example2:storage.disk3.sys3.xyz     192.168.2.15:3260     ONLINE    ␣
↪disk_5
iqn.2001-04.com.example2:storage.disk4.sys3.xyz     192.168.2.15:3260     ONLINE    ␣
↪disk_6
iqn.2001-04.com.example2:storage.disk5.sys3.xyz     192.168.2.15:3260     ONLINE    ␣
↪disk_7
```

Show the default value for target attributes.

```
Storage> iscsi target attr showdefault

Attribute                Value
---------                -----
replacement_timeout      122
```

```
noop_timeout              5
noop_interval             13
login_timeout             10
logout_timeout            15
cmds_max                  128
queue_depth               32
```

Show the value for target attributes of all known targets.

```
Storage> iscsi target attr showall

Attribute                 Value      Target
---------                 -----      ------
replacement_timeout       123        iqn.1992-08.com.iscsi:sn.84268871
noop_timeout              5          iqn.1992-08.com.iscsi:sn.84268871
noop_interval             121        iqn.1992-08.com.iscsi:sn.84268871
login_timeout             10         iqn.1992-08.com.iscsi:sn.84268871
logout_timeout            15         iqn.1992-08.com.iscsi:sn.84268871
cmds_max                  128        iqn.1992-08.com.iscsi:sn.84268871
queue_depth               32         iqn.1992-08.com.iscsi:sn.84268871
replacement_timeout       124        iqn.2009-01.com.example:storage.disk0.lun0
noop_timeout              5          iqn.2009-01.com.example:storage.disk0.lun0
noop_interval             121        iqn.2009-01.com.example:storage.disk0.lun0
login_timeout             10         iqn.2009-01.com.example:storage.disk0.lun0
logout_timeout            15         iqn.2009-01.com.example:storage.disk0.lun0
cmds_max                  128        iqn.2009-01.com.example:storage.disk0.lun0
queue_depth               32         iqn.2009-01.com.example:storage.disk0.lun0
```

Show the value for target attributes of a given target.

```
Storage> iscsi target attr show iqn.1992-08.com.iscsi:sn.84268871

Attribute                 Value
---------                 -----
replacement_timeout       123
noop_timeout              5
noop_interval             121
login_timeout             10
logout_timeout            15
cmds_max                  128
queue_depth               32
```

Set the default value of login_timeout to 10 seconds.

```
Storage> iscsi target attr setdefault login_timeout 10
Sucesss.
```

Set the value of logout_timeout for all known targets to 20 seconds.

```
Storage> iscsi target attr setall logout_timeout 20
Changes would be applicable after next login into the target.
Sucesss.
```

Set the value of noop_interval for a given target to 30 seconds.

```
Storage> iscsi target attr set iqn.1992-08.com.iscsi:sn.84268871 noop_interval 30
Changes would be applicable after next login into the target.
Sucesss.
```

17.9.5 SEE ALSO

disk(1), hba(1), fencing(1), fs(1), pool(1), scanbus(1), snapshot(1), tier(1), quota(1)

17.10 maxiops

17.10.1 SYNOPSIS

“fs setmaxiops ‘‘ [*fs_name*] *maxiops*

`fs maxiopslist` [*fs_name*]

`fs iopsstat list` [*fs_name*]

`fs iopsstat reset` [*fs_name*]

17.10.2 DESCRIPTION

The MAXIOPS feature enables the user to set/reset maximum I/O operations per second (MAXIOPS) value at the file system level and list/reset the maxiops statistics. When multiple applications use the same storage, it is important to balance the workload between them in a way that allows judicious use of storage resources. Setting MAXIOPS lets you control the number of input-output operations per second that storage under file system can process from an application.

17.10.3 OPTIONS

“fs setmaxiops ‘‘ [*fs_name*] *maxiops*

set/reset MAXIOPS on a file system

`fs maxiopslist`

List the value of maximum I/O operations per second (MAXIOPS) set on all the file systems.

`fs iopsstat list` [*fs_name*]

Display a statistical view of the maximum I/O operations per second (MAXIOPS) for a file system.

`fs iopsstat reset` [*fs_name*]

Reset the MAXIOPS satisticis instead of printing them.

17.10.4 EXAMPLES

Set maximum I/O operations per second (MAXIOPS) value for a file system `fs1` to 1000 :

```
Storage> fs setmaxiops fs1 1000
```

Reset maximum I/O operations per second (MAXIOPS) value for a file system `fs1` :

```
Storage> fs setmaxiops fs1 0
```

List maximum I/O operations per second(MAXIOPS) values for all file systems :

```
Storage> fs maxiopslist

    FS                        MAXIOPS
    ========================= =========
    fs1                                   10000
    fs2                                 40000
```

```
    fs3                                 0
    fs4                                                 2000
```

Display statistical view of the maximum I/O operations per second (MAXIOPS) for a file system:

```
Storage> fs iopsstat list fs1
    Listing IOPS stats for every 5 seconds for fs1.....
    Press [CTRL+C] to stop!...
    Mon Jul 10 18:14:36 IST 2017
    Stats for Node test-clus_01
                                        AVG PER SECOND VOLUMEGROUP STATISTICS
    TYP NAME                            MaxIOPS     IncomingIOPS    ServicedIOPS
↪     QueuedIOPS(Transient)
    grp fs1_volgrp                        20000                0               0
↪             0


    Stats for Node test-clus_02
                                        AVG PER SECOND VOLUMEGROUP STATISTICS
    TYP NAME                            MaxIOPS     IncomingIOPS    ServicedIOPS
↪     QueuedIOPS(Transient)
    grp fs1_volgrp                        20000                0               0
↪             0
```

Reset maximum I/O operations per second (MAXIOPS) value for a file system `fs1`:

```
Storage> fs iopsstat reset fs1
    ACCESS fs INFO V-288-0 Resetting IOPS stats for fs1.....
    ACCESS fs SUCCESS V-288-0 Sucessfully resetted the IOPS stat on all nodes
```

17.10.5 SEE ALSO

disk(1), hba(1), fencing(1), fs(1), pool(1), scanbus(1), snapshot(1), tier(1), iscsi(1), cifs(1)

17.11 pool

17.11.1 SYNOPSIS

`pool create` *pool_name disk1* [, *disk2*,...] [*isolated={yes|no}*]

`pool adddisk` *pool_name disk1* [, *disk2*,...]

`pool destroy` *pool_name*

`pool free` [*pool_name*]

`pool list`

`pool markdiskspare` *pool_name disk1* [, *disk2*,...]

`pool removedisk spare` *pool_name disk1* [, *disk2*,...]

`pool mvdisk` *src_pool dest_pool disk1* [, *disk2*,...]

`pool rename` *old_name new_name*

`pool rmdisk` *disk1* [, *disk2*,...]

17.11.2 DESCRIPTION

The storage `pool` command manage logical storage pools.

17.11.3 OPTIONS

`pool create` ***pool_name disk1* [, *disk2*,...] [*isolated={yes|no}*]** Create a logical pool from a given set of disks. If isolated=yes, then a new disk group is created. The default value for isolated is isolated=no.

`pool adddisk` ***pool_name disk1* [, *disk2*,...]** Add a set of disks to a logical pool. If the disk(s) are resolved only by removing/rewriting the partition table of the disk, use `Storage>disk format` *disk1*.

`pool destroy` ***pool_name*** Destroy a pool.

`pool free` **[*pool_name*]** List free space information.

`pool list` List all the pools.

:`pool markdiskspare pool_name` ***disk1* [, *disk2*,...]** Mark a disk as a spare disk and add it to a pool which is later used for hot-relocation. In case of failure of a disk or a plex, the affected subdisks are relocated to disks designated as spare disks.

:`pool removedisk spare pool_name` ***disk1* [, *disk2*,...]** Remove the spare disk flag set on a disk by `markdiskspare` command.

`pool mvdisk` ***src_pool dest_pool disk1* [, *disk2*,...]** Move disks from one pool to another.

`pool rename` ***old_name new_name*** Rename a pool.

`pool rmdisk` ***disk1* [, *disk2*,...]** Remove disks from a pool.

17.11.4 EXAMPLES

Create a pool `pool1` with disks `Disk_0`, `Disk_1`.

```
Storage> pool create pool1 Disk_0,Disk_1 isolated=yes
ACCESS pool Success V-288-1015 Pool pool1 created successfully.
100% [#] Creating pool pool1
```

View the list of pools.

```
Storage> pool list

Pool  Pool Type  List of disks
=====  ========  ==============
pool1  Isolated  Disk_0 Disk_1
pool2  Normal    Disk_2 Disk_5
```

Destroy pool `pool1`.

```
Storage> pool destroy pool1
ACCESS pool SUCCESS V-288-2056 successfully destroyed the pool.
```

Remove disks from a pool.

```
Storage> pool rmdisk Disk_2
ACCESS pool Success V-288-1360 Disk(s) Disk_2 have been removed successfully.
```

Add disk `Disk_2` to pool `pool2`.

```
Storage> pool adddisk pool2 Disk_2
ACCESS pool Success V-288-1001 Disk(s) Disk_2 are added to pool2 successfully.
```

Mark disk Disk_2 in `pool2` as spare disk.

```
Storage> pool markdiskspare pool_name disk1[,disk2,...]
Storage> pool markdiskspare pool2 Disk_2
ACCESS Pool SUCCESS V-288-1360 Disk(s) Disk_2 have been marked as spared disks␣
↪successfully.
```

Remove spare disk flag on Disk_2 in `pool2`.

```
Storage> pool removediskspare pool_name disk1[,disk2,...]
Storage> pool removediskspare pool2 Disk_2
ACCESS Pool SUCCESS V-288-0 Disk(s) Disk_2 have been removed as spared disks␣
↪successfully.
```

Rename `pool1` to `p01`.

```
Storage> pool rename pool1 p01
ACCESS pool Success V-288-1017 Pool rename successful.
```

Move disk `Disk_0` from pool `p01` to pool `pool2`.

```
Storage> pool mvdisk p01 pool2 Disk_0
ACCESS pool Success V-288-1002 Disk(s) moved successfully.
```

View the free space in each of the pools.

```
Storage> pool free
Pool                                    Free Space     Total Space    Use%
==================================      ==========     ===========    ====
p01                                     989.64M         989.64M        0%
pool2                                   2.90G           2.90G          0%
```

View the free space in `pool2`.

```
Storage> pool free pool2
Disk             Free Space     Total Space    Use%
====             ==========     ===========    ====
Disk_0           989.64M        989.64M          0%
Disk_2           991.69M        991.69M          0%
Disk_5           991.69M        991.69M          0%
```

17.11.5 SEE ALSO

disk(1), hba(1), fencing(1), fs(1), scanbus(1), snapshot(1), tier(1), iscsi(1), quota(1)

17.12 quota

17.12.1 SYNOPSIS

`quota fs enable` [*fs_name*] [`userquota`|`groupquota`]

`quota fs disable` [*fs_name*] [`userquota`|`groupquota`]

`quota fs status` [*fs_name*] [`userquota`|`groupquota`]

`quota fs set` `userquota`|`groupquota` *user_or_group_names domain_name* [`hardlimit`|`softlimit`] [`numinodes`|`numspace`] [*value*] [*fs_name*]

`quota fs setall` `userquota` | `groupquota` [`hardlimit` | `softlimit`] [`numinodes` | `numspace`] [*value*] [*fs_name*]

`quota fs setbygroup` *group_names domain_name* [`hardlimit`|`softlimit`] [`numinodes`|`numspace`] [*value*] [*fs_name*]

`quota fs show` [*fs_name*] [`userquota`|`groupquota`] [*user_or_group_names*] [*domain_name*]

`quota fs setdefault` `userquota`|`groupquota` `hardlimit`|`softlimit` `numinodes`|`numspace` [*value*] [*fs_name*]

`quota fs showdefault` [*fs_name*] [`userquota`|`groupquota`]

`quota cifshomedir enable` [`userquota` | `groupquota`] `quota cifshomedir disable` [`userquota` | `groupquota`] `quota cifshomedir status` [`userquota` | `groupquota`] `quota cifshomedir set` `userquota` | `groupquota` *user_or_group_names domain_name* [`hardlimit` | `softlimit`] [`numinodes` | `numspace`] [*value*] `quota cifshomedir setall` `userquota` | `groupquota` [`hardlimit` | `softlimit`] [`numinodes` | `numspace`] [*value*] `quota cifshomedir show` [`userquota` | `groupquota`] [*user_or_group_names*] [*domain_name*] `quota cifshomedir showdetail` [`userquota`|`groupquota`] [*user_or_group_names*] [*domain_name*] `quota cifshomedir setdefault` `userquota` | `groupquota` `hardlimit` | `softlimit` `numinodes` | `numspace` [*value*] `quota cifshomedir showdefault` [`userquota`|`groupquota`]

17.12.2 DESCRIPTION

The `storage quota` commands are for configuring disk quotas on file systems for users and groups. There are two types of disk quotas, one is a usage quota (`numspace`), and the other is an inode quota (`numinodes`). A quota limit can be set as a soft quota limit (`soflimit`) where users are warned against exceeding the quota limits, and there is a grace period during which the user is allowed to exceed the quota limits. After the grace period is over, the user is not allowed to exceed the quota limits. Hard quota limits (`hardlimit`) can also be set so that the user is not allowed to exceed the quota limits. The `soflimit` has to be less than the `hardlimit` for any type of quota.

The `storage quota fs` commands are for configuring quotas on file systems that are not part of the CIFS home directories. The `storage quota cifshomedir` commands are for configuring quotas on CIFS home directories. All users and groups visible through different sources of Name Service Lookup (nsswitch), such as, local users, LDAP, NIS, Windows users, and so on, can be configured for quotas.

17.12.3 OPTIONS

`quota fs enable` [*fs_name*] [`userquota`|`groupquota`]

`quota cifshomedir enable` [`userquota`|`groupquota`]

Enable user or group quota on a file system or CIFS home directories.

`quota fs disable` [*fs_name*] [`userquota`|`groupquota`]

`quota cifshomedir disable` [`userquota`|`groupquota`

Disable user or group quota on a file system or CIFS home directories.

`quota fs status` [*fs_name*] [`userquota`|`groupquota`]

`quota cifshomedir status` [`userquota`|`groupquota`]

Show the status of quota settings on a file system or CIFS home directories. This only shows if the quota is enabled or disabled.

`quota fs set` `userquota`|`groupquota` *user_or_group_names domain_name* [`hardlimit`|`softlimit`] [`numinodes`|`numspace`] [*value*] [*fs_name*]

`quota cifshomedir set` `userquota`|`groupquota` *user_or_group_names domain_name* [`hardlimit`|`softlimit`] [`numinodes`|`numspace`] [*value*]

Set a quota value for users or groups on a file system or CIFS home directories. If *value* is not specified, then the default value set from the respective `setdefault` commands is used to configure the quota limit. If *value* is 0, it is treated as unlimited quota. If all values of user or group quota are 0, the user or group is automatically deleted from the quota settings, which means `quota fs status` does not show this user's or group's settings, as all quota values are unlimited for it.

`quota fs setall` `userquota`|`groupquota` [`hardlimit`|`softlimit`] [`numinodes`|`numspace`] [*value*] [*fs_name*]

`quota cifshomedir setall` `userquota`|`groupquota` [`hardlimit`|`softlimit`] [`numinodes`|`numspace`] [*value*]

Set quota value for all users and groups for whom the quota has already been set with `set` commands. Other users and groups (for whom quota has not been set previously) are not affected. If *value* is not specified, then the default value set from the respective `setdefault` commands is used to configure the quota limit. If *value* is 0, it is treated as an unlimited quota. If all values of user or group quota are 0, the user or group are automatically deleted from the quota settings, which means `quota fs status` does not show this user's or group's settings, as all quota values are unlimited for it.

`quota fs setbygroup` *group_names domain_name* [`hardlimit`|`softlimit`] [`numinodes`|`numspace`] [*value*] [*fs_name*]

Set the user quota for users of specified groups.

`quota fs show` [*fs_name*] [`userquota`|`groupquota`] [*user_or_group_names*] [*domain_name*]

`quota cifshomedir show` [`userquota`|`groupquota`] [*user_or_group_names*] [*domain_name*]

`quota cifshomedir showdetail` [`userquota`|`groupquota`] [*user_or_group_names*] [*domain_name*]

Show the quota values that are already set. This also shows the consumed (used space) quota of users and groups. `cifshomedir show` shows the general quota values on the CIFS home directories. `cifshomedir showdetail` shows the detailed quota values set on each file system for CIFS home directories.

`quota fs setdefault` `userquota`|`groupquota` `hardlimit`|`softlimit` `numinodes`|`numspace` [*value*] [*fs_name*]

`quota cifshomedir setdefault` `userquota`|`groupquota` `hardlimit`|`softlimit` `numinodes`|`numspace` [*value*]

Set the default value that is used for quota limits. The values are put in a configuration file only. The actual application of quotas can be done with `set` and `setall` commands using these default values.

`quota fs showdefault` [*fs_name*] [`userquota|groupquota`]

`quota fs showdefault` [`userquota|groupquota`]

Show the default quota values from the configuration file.

17.12.4 EXAMPLES

Enable quota (user and group quota) for file system `fs1`:

```
Storage> quota fs enable fs1
OK Completed
```

Enable quota (user and group quota) for CIFS home directories:

```
Storage> quota cifshomedir enable
OK Completed
```

Disable userquota for file system `fs1`:

```
Storage> quota fs disable fs1 userquota
OK Completed
```

Disable groupquota for CIFS home directories:

```
Storage> quota cifshomedir disable groupquota
OK Completed
```

Show status of file system quota (enabled or disabled):

```
Storage> quota fs status

FS name                    User Quota   Group Quota
=======                    ==========   ==========
fsmirror                   Disabled     Disabled
quotafs                    Enabled      Enabled
striped1                   Enabled      Enabled
fs1                        Disabled     Enabled
OK Completed
```

Show status of CIFS home directory quota (enabled or disabled):

```
Storage> quota cifshomedir status

FS name                    User Quota   Group Quota
=======                    ==========   ==========
CIFS homedirectories       Enabled      Disabled
OK Completed
```

Set userquota (hardlimit and numinodes) of user qtuser on file system `fs1`:

```
Storage> quota fs set userquota qtuser qtdomain hardlimit numinodes 957 fs1
OK Completed
Storage> quota fs show fs1 userquota qtuser qtdomain
User quota details for file system fs1:

User Name        Space Used      Soft Space      Hard Space      Inodes Used     Soft␣
↪Inodes     Hard Inodes
```

```
==========      ==========      ==========      ==========      ==========
↪==========      ==========
qtdomain\\qtuser 0               0               0               0               0
↪           957
OK Completed
```

Set user quota (hardlimit and numinodes) of user qtuser on CIFS home directories:

```
Storage> quota cifshomedir set userquota qtuser qtdomain hardlimit numinodes 6549
OK Completed
Storage> quota cifshomedir show

User quota details for CIFS home directories:

User Name       Space Used      Soft Space      Hard Space      Inodes Used     Soft
↪Inodes     Hard Inodes
==========      ==========      ==========      ==========      ==========
↪==========      ==========
qtdomain\\qtuser 0               20M             100M            0               1000
↪           6549
OK Completed
```

Set all existing user quotas to default values:

```
Storage> quota fs show fs1
User quota details for file system fs1:
User Name       Space Used      Soft Space      Hard Space      Inodes Used     Soft
↪Inodes     Hard Inodes
==========      ==========      ==========      ==========      ==========
↪==========      ==========
a1              0               0               10G             0               1000
↪          10000
qtuser          0               0               0               0               0
↪          957
qtuser2         0               1000K           0               0               0
↪          0

Storage> quota fs setall userquota
OK Completed

Storage> quota fs show fs1

User quota details for file system fs1:

User Name       Space Used      Soft Space      Hard Space      Inodes Used     Soft
↪Inodes     Hard Inodes
==========      ==========      ==========      ==========      ==========
↪==========      ==========
a1              0               0               10G             0               1000
↪          1000
qtuser          0               0               0               0               0
↪          1000
qtuser2         0               1000K           0               0               0
↪          1000
```

Set all existing user quotas for CIFS home directories:

```
Storage> quota cifshomedir show

User Name        Space Used      Soft Space      Hard Space      Inodes Used     Soft␣
↪Inodes     Hard Inodes
==========       ==========      ==========      ==========      ==========      ␣
↪==========       ==========
qtuser           0               20M             100M            0               1000 ␣
↪           6549
OK Completed
Storage> quota cifshomedir setall userquota softlimit numinodes 198
OK Completed
Storage> quota cifshomedir show

User quota details for CIFS home directories:

User Name        Space Used      Soft Space      Hard Space      Inodes Used     Soft␣
↪Inodes     Hard Inodes
==========       ==========      ==========      ==========      ==========      ␣
↪==========       ==========
qtuser           0               20M             100M            0               198  ␣
↪           6549
OK Completed
```

Set user quotas for users of specified groups.

```
Storage> quota fs setbygroup cifsgroup1 local softlimit numspace 50M fs1
OK Completed
Storage> quota fs show

User quota details for file system fs1:

User Name        Space Used      Soft Space      Hard Space      Inodes Used     Soft␣
↪Inodes     Hard Inodes
==========       ==========      ==========      ==========      ==========      ␣
↪==========       ==========
cifsuser1        0               50M             0               0               0    ␣
↪           0
cifsuser2        0               50M             0               0               0    ␣
↪           0
OK Completed
```

Show file system quota values:

```
Storage> quota fs show

User quota details for file system quotafs:

User Name        Space Used      Soft Space      Hard Space      Inodes Used     Soft␣
↪Inodes     Hard Inodes
==========       ==========      ==========      ==========      ==========      ␣
↪==========       ==========
quotauser        10M             1M              20M             1               5    ␣
↪           1000
quotauser        9M              1M              10M             1               0    ␣
↪           0
qtuser           10M             10M             20M             9               5    ␣
↪           1000
qtuser2          19M             5M              20M             1               0    ␣
↪           1000
```

```
User quota details for file system fs1:

User Name        Space Used      Soft Space      Hard Space      Inodes Used     Soft␣
↪Inodes     Hard Inodes
==========       ==========      ==========      ==========      ==========      ␣
↪==========       ==========
a1               0               0               10G             0               1000 ␣
↪          1000
qtuser           0               0               0               0               0    ␣
↪          1000
qtuser2          0               1000K           0               0               0    ␣
↪          1000

User quota details for file system longfilesystemnameforqt:

User Name        Space Used      Soft Space      Hard Space      Inodes Used     Soft␣
↪Inodes     Hard Inodes
==========       ==========      ==========      ==========      ==========      ␣
↪==========       ==========
qtuser           0               0               0               0               901  ␣
↪          1000
OK Completed
```

Show CIFS home directory quota values:

```
Storage> quota cifshomedir show

User quota details for CIFS home directories:

User Name        Space Used      Soft Space      Hard Space      Inodes Used     Soft␣
↪Inodes     Hard Inodes
==========       ==========      ==========      ==========      ==========      ␣
↪==========       ==========
qtuser           0               20M             100M            0               198  ␣
↪          6549
OK Completed
```

Set default group quota value:

```
Storage> quota fs setdefault groupquota hardlimit numspace 1T
OK Completed
```

Set default CIFS home directory user quota value:

```
Storage> quota cifshomedir setdefault userquota hardlimit numspace 2T
OK Completed
```

Show default quota values:

```
Storage> quota fs showdefault

Default quota values:
=====================

Title                User/Group Quota     Soft Space      Hard Space      Soft Inodes␣
↪     Hard Inodes
=====                ================     ==========      ==========      ========== ␣
↪     ==========
```

```
Default Quota        User Quota           -               -               -
↪    1000
Default Quota        Group Quota          -               1T              -
↪    -


Per FS default quota values:
============================

FS Name              User/Group Quota     Soft Space      Hard Space      Soft Inodes
↪    Hard Inodes
========             ================     ==========      ==========      ==========
↪    ==========
fs1                  User Quota           -               -               -
↪    1000
OK Completed
```

Show default CIFS home directory quota values:

```
Storage> quota cifshomedir showdefault

CIFS homedir default quota values:
==================================

User/Group Quota     Soft Space      Hard Space      Soft Inodes     Hard Inodes
================     ==========      ==========      ==========      ==========
User Quota           -               2T              -               -
Group Quota          -               -               -               -
OK Completed
```

17.12.5 SEE ALSO

disk(1), hba(1), fencing(1), fs(1), pool(1), scanbus(1), snapshot(1), tier(1), iscsi(1), cifs(1)

17.13 rollback

17.13.1 SYNOPSIS

`rollback create space-optimized` *rollback_name fs_name* [cacheobj]

`rollback create full-sized` *rollback_name fs_name pool*

`rollback restore` *fs_name rollback_name*

`rollback refresh` *rollback_name fs_name*

`rollback destroy` *rollback_name fs_name*

`rollback list` [fs_name/rollback_name]

`rollback online` *rollback_name fs_name*

`rollback offline` *rollback_name fs_name*

`rollback cache create` *cache_name* [cache_size] [pool]

`rollback cache destroy` *cache_name*

`rollback cache list` [cache_name]

`rollback cache growby` *cache_name cache_size*

`rollback cache growto` *cache_name cache_size*

17.13.2 DESCRIPTION

The storage `rollback` commands manage volume-level snapshots. All rollback commands take a file system as an argument and perform operations on the underlying volume of that file system.

Both space-optimized and full-sized rollbacks are supported. Space-optimized rollbacks use a storage cache and do not need a complete copy of the original volume's storage space. However, space-optimized rollbacks are not suitable for write-intensive volumes, because the copy-on-wirte mechanism may degrade the performance of the volume. Full-sized rollbacks use more storage, but it has little impact on write performance after synchronization is completed.

The `rollback` command can be used to perform the following operations:

- Creating/destroying/listing instant rollbacks for a given file system.
- Restoring a file system by a given instant rollback.
- Refreshing an instant rollback from a file system.
- Onlining/offlining instant rollbacks for a given file system.
- Creating/destroying/listing cache object that can be used for instant rollbacks.

17.13.3 OPTIONS

`rollback create space-optimized` *rollback_name fs_name* [cacheobj] Create a space-optimized rollback for a specified file system. If the cacheobj is specified, then the shared cache object is used. Or the system automatically creates a cache object for the rollback.

`rollback create full-sized` *rollback_name fs_name pool* Create a full-sized rollback for a specified file system. The disks used for the rollback are allocated from the specified pool.

`rollback restore` *fs_name rollback_name* Restore a file system by a given rollback.

rollback refresh *rollback_name fs_name* Refresh an instant rollback from a file system.

rollback destroy *rollback_name fs_name* Destroy the instant rollback of a file system.

rollback list [fs_name/rollback_name] Display all the instant rollbacks of the specified file system. If rollback name is specified, then display information about the specifed rollback. If no file system name or rollback name is specified, then instant rollbacks of all the file systems are displayed.

rollback online *rollback_name fs_name* Place the instant rollback online.

rollback offline *rollback_name fs_name* Place the instant rollback offline.

rollback cache create *cache_name* **[cache_size] [pool]** Create a shared cache object or convert the file system to a shared cache object for space-optimized rollbacks for the file system. If cache_size and pool are both specified, create a shared cache object. If only cache_name is specified, convert the file system to a shared cache object. cache_name should be the same as an existing file system name that is to be converted. There is a confirmation message in the Veritas Access CLI asking if you want to convert the specified file system to a cache object.

rollback cache destroy *cache_name* Destroy the shared cache object.

rollback cache list [cache_name] Display the shared cache objects. If no cache_name is specified, all cache objects are displayed. The disabled cache object is listed with '-' as the attribute. If cache_name is specified and the cache object is disabled, the cache object is enabled automatically.

rollback cache growby *cache_name cache_size* Grow the size of the cache object by a specified amount.

rollback cache growto *cache_name cache_size* Grow the size of the cache object to a specified amount.

17.13.4 EXAMPLES

Create a space-optimized rollback of a file system.

```
Storage> rollback create space-optimized snap4 fs4
100% [#] Create rollback
```

Create a full-sized rollback of a file system.

```
Storage> rollback create full-sized snap5 fs4 pool1
100% [#] Create rollback
```

Restore a file system by a given instant rollback.

```
Storage> rollback restore fs4 snap4
```

Destroy the instant rollback of a file system.

```
Storage> rollback destroy snap4 fs4
100% [#] Destroy rollback
```

View the list of instant rollbacks.

```
Storage> rollback list
NAME                    TYPE          FILESYSTEM                SNAPDATE
roll5                   fullinst      fs4                       2010/10/15 20:04
roll1                   spaceopt      bigfs                     2010/10/15 17:03
```

View the list of instant rollbacks.

```
Storage> rollback list fs4
NAME                    TYPE           SNAPDATE            CHANGED_DATA    SYNCED_DATA
roll5                   fullinst       2010/10/15 20:04    640K(0.1%)      800M(100%)
```

Create a shared cache object.

```
Storage> rollback cache create cobj1 100m pool1
100% [#]
```

Convert the file sytem to a shared cache object.

```
Storage> rollback cache create fs_to_cache
ACCESS rollback WARNING V-288-0 Filesystem fs_to_cache will be converted to cache␣
↪object. All data on Filesystem fs_to_cache will be lost
ACCESS rollback WARNING V-288-0 Are you sure to convert fs_to_cache to cache object?␣
↪(yes/no)
yes
100% [#]
```

Grow the size of the cache object by a specified amount. In following example, the cache size is grown by 1M.

```
Storage> rollback cache growby cobj1 1M
ACCESS rollback SUCCESS V-288-0 Size of cache object cobj1 extended successfully.
```

Grow the size of the cache object to a specified amount. In following example, the cache size is grown to 10G.

```
Storage> rollback cache growto cobj1 10G
ACCESS rollback SUCCESS V-288-0 Size of cache object cobj1 extended successfully.
```

Display the shared cache object.

```
Storage> rollback cache list
CACHE NAME                      TOTAL(Mb)   USED(Mb)  (%)    AVAIL(Mb)  (%)    SDCNT
fs_cache1                              10          4  (40)          6  (60)       0
cache2                                  -          -    -           -    -        -

Storage> rollback cache list cache2

rollbacks located on cache cache2:
roll2


Storage> rollback cache list cache_disabled

rollbacks located on cache cache_disabled:
roll3

ACCESS rollback WARNING V-288-0 Cache object cache_disabled was DISABLED, trying to␣
↪restart it.
ACCESS rollback INFO V-288-0 Cache object cache_disabled started successfully.
```

17.13.5 SEE ALSO

disk(1), hba(1), fencing(1), fs(1), pool(1), scanbus(1), tier(1), iscsi(1), quota(1)

17.14 scanbus

17.14.1 SYNOPSIS

`scanbus` [force]

17.14.2 DESCRIPTION

The storage `scanbus` command scans all the SCSI devices connected to all the nodes of the cluster. It scans the disks on all the nodes without interrupting existing I/O activity, and it updates the configuration. It does not inform the user even if there is a change in the storage configuration. Users can see the latest storage configuration with the `disk list` command.

17.14.3 OPTIONS

force Tries to import pools forcefully. This may help when normal `scanbus` alone fails.

17.14.4 EXAMPLES

Scan SCSI Bus for newly added disks.

```
    Storage> scanbus
100% [#] Scanning the bus for disks
```

17.14.5 SEE ALSO

disk(1), hba(1), fencing(1), fs(1), pool(1), snapshot(1), tier(1), iscsi(1), quota(1)

17.15 snapshot

17.15.1 SYNOPSIS

`snapshot create` *snapshot_name fs_name* [removable=yes|removable=no]

`snapshot restore` *snapshot_name fs_name*

`snapshot destroy` *snapshot_name fs_name*

`snapshot list` [fs_name] [schedule_name]

`snapshot online` *snapshot_name fs_name*

`snapshot offline` *snapshot_name fs_name*

`snapshot quota on` *fs_name* [capacity_limit]

`snapshot quota off` [fs_name] [remove_limit]

`snapshot quota list`

`snapshot schedule create` *schedule_name fs_name max_snapshot_limit minute* [hour]

[day_of_the_month] [month] [day_of_the_week] [max_num_of_no_named_fset]

`snapshot schedule modify` *schedule_name fs_name max_snapshot_limit minute* [hour]

[day_of_the_month] [month] [day_of_the_week] [max_num_of_no_named_fset]

`snapshot schedule destroyall` *schedule_name fs_name*

`snapshot schedule preserve` *schedule_name fs_name snapshot_name*

`snapshot schedule show` *fs_name* [schedule_name]

`snapshot schedule delete` *fs_name* [schedule_name]

17.15.2 DESCRIPTION

The storage `snapshot` commands manage file system level snapshots. All snapshot commands take a file system as an argument and perform operations on that file system.

The `snapshot` command can be used to perform the following:

- Adding/removing/destroying/listing snapshots for a given file system.
- Restoring a file system by a given snapshot.
- Onlining/offlining snapshots for a given file system.
- Turning on/off quota value for the space that can be utilized for snapshot creation corresponding to a given file system name.
- Creating/modifying a schedule that automatically creates snapshots for a given file system every X hours and Y minutes.
- Destroying all automated snapshots corresponding to a given schedule name and file system name. If any snapshot is preserved or online, the command fails.
- Preserving an automated snapshot so that it is not automatically removed once the command `snapshot schedule autoremove` has been run.
- Displaying/deleting schedules created for automated snapshot creation and removal.

Automated snapshot creation

A snapshot can be created in two ways: either manually using the `snapshot create` command, or by creating a schedule that calls the `snapshot create` command periodically depending on the values entered for the number of hours or minutes after which this command should be run. (**Notice**: We suggest to only create one schedule for a specified file system, otherwise `storage snapshot schedule destroyall` may take a long time).

Thus a key feature of the `snapshot` command is to allow for creation of a schedule that can create a snapshot in an automated manner. This is done by storing the values for minute, hour, day-of-month, month, and day-of-week in the crontab along with the name of the file system for which the snapshot is created automatically. To distinguish the automated snapshots, a timestamp corresponding to their time of creation is appended to the schedule name. Thus a snapshot created under schedule1 on 27th February 2009 at 11 am is named as:

```
schedule1_Feb_27_2009_11_00_01_IST
```

The following are the main parameters and the type of values one can provide while creating or modifying a schedule for automated snapshot creation:

1. `Schedule name:` This specifies the name of the schedule corresponding to which a snapshot is created automatically. The schedule name cannot contain an underscore '_' as part of its input value. This is by design. So a schedule name such as s_1 is not allowed.

2. `Max Snapshot Limit:` This specifies the number of snapshots that can be created for a given file system and schedule name. This field accepts numeric input only. The range of this value is from '1' to '366'. This value would imply that only x number of snapshots can be created for a given file system and schedule name. If the number of snapshots corresponding to a schedule name is equal to or greater than the value of this field, then snapshots are automatically destroyed till the number of snapshots is less than the maximum snapshot limit value.

3. `Minute:` This field may contain either an asterisk like *'/15'', which implies every 15 minutes. (Notice: If using* '/xx' format, the smallest value for 'xx' is 15) or a numeric value between 0-59.

4. `Hour:` This field may contain either an asterisk '*', which implies every hour, or a numeric value between 0-23.

5. `Day of month:` This field may contain either an asterisk '*', which implies every day of the month, or a numeric value between 1-31.

6. `Month:` This field may contain either an asterisk '*', which implies run every month, or a numeric value between 1-12. In addition to the numeric values, this field can also accept names of month as arguments, with the first three letters of the month (all in lowercase) serving as input for the given parameter.

7. `Day of Week:` This field may contain either an asterisk '*', which implies every day of the week, or a numeric value between 0-6, with 0 being interpreted as Sunday, 1 as Monday and so on. In addition, this paramter can also accept names, with the first three letters of the month (all in lowercase) serving as input values.

8. `Number of maximum removing no named fset:` This field has a default value of 2, which implies that if the current removing no-named-fset exceeds that value, succssive snapshot auto-create can be skipped.

Note:

1. By default, the parameters hour, day-of-month, month, and day-of-week for `snapshot schedule create` command contain a '*' or an asterisk value, while max_num_of_no_named_fset is '2' by default as they are all optional arguments.
2. The parameters minute, hour, day-of-month, month, and day-of-week for `snapshot schedule create` and `snapshot schedule modify` can accept numeric values in the form of ranges, where a range is defined as two numbers separated by a hyphen. So if one wishes to run the schedule between 1 am and 4 am, then one can specify a value 1-4 for 'hour' parameter, with the range being inclusive.
3. Similarly, one can specify a step value for other parameters, that is, day-of-month, month, and day-of-week as well. Step values are also allowed after an asterisk '*', so if one wishes to run a schedule every two hours, one just has to specify '*/2' as the input value for the 'hour' parameter.

4. The range of value `max_num_of_no_named_fset` is from 2 to 5.

So to create a snapshot every two and half hours with at most 50 snapshots per schedule name with a maximum of 4 removing no-named-fsets, you have to run the following command:

```
snapshot schedule create sched1 fs1 50 */30 */2 * * * 4
```

17.15.3 OPTIONS

`snapshot create` *snapshot_name fs_name* [removable=yes|removable=no]

Create a snapshot for a specified file system. If the removable attribute is `yes` and if it is offline, then it is removed automatically, if the file system runs out of space.

`snapshot restore` *snapshot_name fs_name*

Restore a file system by a given snapshot.

`snapshot destroy` *snapshot_name fs_name*

Destroy the snapshot of a file system.

`snapshot list` [fs_name] [schedule_name]

Display all the snapshots of the specified file system and schedule name. If no file system is specified, snapshots of all the file systems are displayed, whereas if no schedule_name is specified, then snapshots created under fs_name are displayed.

`snapshot online` *snapshot_name fs_name*

Place the snapshot online.

`snapshot offline` *snapshot_name fs_name*

Place the snapshot offline.

`snapshot quota on` *fs_name* [capacity_limit]

Disallow creation of snapshots on the given file system when the space used by all the snapshots of that file system exceeds a given capacity limit.

`snapshot quota off` [fs_name] [remove_limit]

Disable quota capacity limit for a specified system. The default remove_limit value is true. If it is false, the capacity limit is not reset.

`snapshot quota list`

Display snapshot quota information of all the file systems.

`snapshot schedule create` *schedule_name fs_name max_snapshot_limit minute* [hour] [day_of_the_month] [month] [day_of_the_week] [max_num_of_no_named_fset]

Create a schedule for automated snapshot creation of a particular file system.

`snapshot schedule modify` *schedule_name fs_name max_snapshot_limit minute* [hour] [day_of_the_month] [month] [day_of_the_week] [max_num_of_no_named_fset]

Modify the schedule for automated snapshot creation of a particular file system.

`snapshot schedule destroyall` *schedule_name fs_name*

Destroy all automated snapshots corresponding to a given schedule name and file system name. If any snapshot is preserved or online, the command fails.

`snapshot schedule preserve` *schedule_name fs_name snapshot_name*

Preserve a limited number of snapshots corresponding to an existing schedule and a specific file system name so that they are not removed as part of the snapshot schedule autoremove command.

`snapshot schedule show` *fs_name* [schedule_name]

Show all schedules that have been set for automated snapshot creation.

`snapshot schedule delete` *fs_name* [schedule_name]

Delete the schedule set for automated snapshot creation for a particular file system or for a particular schedule.

17.15.4 EXAMPLES

Create a snapshot of a file system.

```
Storage> snapshot create snapshot1 fs1
100% [#] Create snapshot
```

Restore a file system by a given snapshot.

```
Storage> snapshot restore snapshot1 fs1
```

Destroy the snapshot of a file system.

```
Storage> snapshot destroy snap1 fs1
100% [#] Destroy snapshot
```

View the list of snapshots.

```
Storage> snapshot list
Snapshot                                FS       Status     ctime                 ␣
↪mtime                  Removable  Preserved  Size
=======                                 ===      ======     =====                 ␣
↪=====                  =========  =========  =======
snap2                                   fs1      offline    2009.Jul.27.02:40:43  ␣
↪2009.Jul.27.02:40:57   no         no         190.0M
sc1_24_Jul_2009_21_34_01_IST            fs1      offline    2009.Jul.24.21:34:03  ␣
↪2009.Jul.24.21:34:03   yes        no         900.0M
sc1_24_Jul_2009_19_34_02_IST            fs1      offline    2009.Jul.24.19:34:04  ␣
↪2009.Jul.24.19:34:04   yes        no         7.0G
presnap_sc1_24_Jul_2009_18_34_02_IST    fs1      offline    2009.Jul.24.18:34:04  ␣
↪2009.Jul.24.18:34:04   yes        yes        125M
sc1_24_Jul_2009_17_34_02_IST            fs1      offline    2009.Jul.24.17:34:04  ␣
↪2009.Jul.24.17:34:04   yes        no         0K
```

Display the list of snapshots for a given file system.

```
Storage> snapshot list fs2
Snapshot                                Schedule Name   Status    ctime                ␣
↪mtime                  Removable   Preserved  Size
======                                  ==============   ======    =====                ␣
↪=====                  =========   =========  =======
snap2                                        -          offline   2009.Jul.27.02:40:43 ␣
↪2009.Jul.27.02:40:57  yes         no         190.0M
sc1_24_Jul_2009_22_34_02_IST                 sc1        offline   2009.Jul.24.22:34:09 ␣
↪2009.Jul.24.22:34:09  yes         no         900.0M
sc1_24_Jul_2009_21_34_01_IST                 sc1        offline   2009.Jul.24.21:34:03 ␣
↪2009.Jul.24.21:34:03  yes         no         7.0G
```

```
sc1_24_Jul_2009_19_34_02_IST             sc1        offline   2009.Jul.24.19:34:04 ␣
↪2009.Jul.24.19:34:04  yes        no         125M
presnap_sc1_24_Jul_2009_18_34_02_IST     -          offline   2009.Jul.24.18:34:04 ␣
↪2009.Jul.24.18:34:04  yes        yes        0K
sc1_24_Jul_2009_17_34_02_IST             sc1        offline   2009.Jul.24.17:34:04 ␣
↪2009.Jul.24.17:34:04  yes        no         97M
```

Display the list of snapshots corresponding to a file system and schedule name.

```
Storage> snapshot list fs1 sc1
Snapshot                                 Status     ctime                      mtime  ␣
↪                  Removable  Size
============================             ======     =====                      =====  ␣
↪                  =========  =======
sc1_24_Jul_2009_22_34_02_IST             offline    2009.Jul.24.22:34:09       2009.
↪Jul.24.22:34:09      yes        190.0M
sc1_24_Jul_2009_21_34_01_IST             offline    2009.Jul.24.21:34:03       2009.
↪Jul.24.21:34:03      yes        900.0M
sc1_24_Jul_2009_20_34_02_IST             offline    2009.Jul.24.20:34:04       2009.
↪Jul.24.20:34:04      yes        7.0G
sc1_24_Jul_2009_19_34_02_IST             offline    2009.Jul.24.19:34:04       2009.
↪Jul.24.19:34:04      yes        125M
sc1_24_Jul_2009_18_34_02_IST             offline    2009.Jul.24.18:34:04       2009.
↪Jul.24.18:34:04      yes        0K
```

Make a snapshot offline.

```
Storage> snapshot offline snapshot1 fs1
100% [#] Offline snapshot
```

Make a snapshot online.

```
Storage> snapshot online snapshot1 fs1
100% [#] Online snapshot
```

Enable the snapshot quota of a file system.

```
Storage> snapshot quota on fs1
Storage> snapshot quota on fs1 1M
```

Disable the snapshot quota of a file system.

```
Storage> snapshot quota off fs2
Storage> snapshot quota off fs3 true
Storage> snapshot quota off fs4 false
Storage> snapshot quota off fs5
```

Display the list of snapshot quotas of all the file systems.

```
Storage> snapshot quota list
FS                                 Quota      Capacity Limit
================================== =====      ==============
fs1                                on               1M
fs2                                off               0
fs3                                off               0
fs4                                off              1M
fs5                                off               0
```

Create a schedule for automated snapshot creation of a given file system every 3 hours on a daily basis and only 30 snapshots can be treated for a given schedule, and the maximum removing no-named-fset is 3.

```
Storage> snapshot schedule create schedule1 fs1 30 * 3 * * * 3
```

Modify the existing schedule so that the snapshot is created every 2 hours on the first day of the week, and only 20 snapshots can be created for a given schedule.

```
Storage> snapshot schedule modify schedule1 fs1 20 * 2 * * 1
```

Destroy all automated snapshots created under a given schedule and file system.

```
Storage> snapshot schedule destroyall schedule1 fs1
```

Preserve a snapshot created according to a given schedule and file system name.

```
Storage> snapshot schedule preserve schedule1 fs1 schedule1_Feb_27_16_42_00_IST
```

List all schedules created for automated snapshot creation corresponding to an existing file system.

```
Storage> snapshot schedule show fs3
FS            Schedule Name              Max Snapshot     Minute     Hour    Day
↪Month   WeekDay
===========   ========================  =============     ======     ====    ===
↪=====   =======
fs3           sched1                          30           */20       *       *
↪*        *
fs3           sched2                          20           */45       *       *
↪*        *
```

List automated snapshot schedules for all file systems.

```
Storage> snapshot schedule show
FS           Schedule Name               Max Snapshot     Minute     Hour    Day
↪Month   WeekDay
===========  ========================  =============     ======     ====    ===
↪=====   =======
fs6          sc1                             10           */50       *       *
↪*        *
fs1          sc1                             10           */25       *       *
↪*        *
```

Delete all schedules created for automated snapshot creation or removal corresponding to an existing file system.

```
Storage> snapshot schedule delete fs1
```

17.15.5 SEE ALSO

disk(1), hba(1), fencing(1), fs(1), pool(1), scanbus(1), tier(1), iscsi(1), quota(1)

17.16 tier

17.16.1 SYNOPSIS

`tier add cloud` *fs_name tier_name service_name region S3|Glacier*

`tier add simple` *fs_name size pool1* [, *disk1*,...]

`tier add mirrored` *fs_name size nmirrors pool1* [, *disk1*,...] [protection=disk|pool]

`tier add striped` *fs_name size ncolumns pool1* [, *disk1*,...] [stripeunit=<kilobytes>]

`tier add mirrored-stripe` *fs_name size nmirrors ncolumns pool1* [, *disk1*,...] [protection=disk|pool] [stripeunit=<kilobytes>]

`tier add striped-mirror` *fs_name size nmirrors ncolumns pool1* [, *disk1*,...] [protection=disk|pool] [stripeunit=<kilobytes>]

`tier addcolumn` *fs_name ncolumns pool1* [, *disk1*,...]

`tier rmcolumn` *fs_name*

`tier remove` *fs_name* [*tier_name*]

`tier list` *fs_name*

`tier move start` *fs_name source_tier destination_tier retrieval_option=Expedited|Standard|Bulk pattern* [atime|mtime] [atime|mtime]

`tier move list`

`tier move status` *JOBID*

`tier move abort` *JOBID*

`tier move dryrun` *fs_name source_tier destination_tier pattern* [atime|mtime] [atime|mtime]

`tier addmirror` *fs_name pool1* [, *disk1*,...][protection=disk|pool]

`tier rmmirror` *fs_name* [*pool_or_disk_name*]

`tier listfiles` *fs_name* {primary|secondary}

`tier mapfile` *fs_name file_path*

`tier policy list`

`tier policy prune list`

`tier policy prune modify` *fs_name delete_after*

`tier policy modify` *fs_name* {primary|secondary} *days minacctemp period*

`tier relocate` *fs_name dirPath*

`tier policy remove` *fs_name*

`tier policy prune remove` *fs_name*

`tier policy run` *fs_name*

`tier schedule list`

`tier schedule modify` *fs_name minute hour day_of_the_month month day_of_the_week node_name*

`tier schedule remove` *fs_name*

`tier query` *fs_name*

`tier allowmetadata yes` *fs_name*

`tier allowmetadata` no *fs_name*

`tier setfastresync` *fs_name pool1* [, *disk1*,...]

`tier unsetfastresync` *fs_name*

`tier stats monitor` *fs_name tier_name* [*interval*]

`tier stats {show|reset|usage}` *fs_name tier_name*

17.16.2 DESCRIPTION

Veritas Access provides two types of tiers, a primary tier and a secondary tier. Each created file system will have only one primary tier initially. This tier cannot be removed. Operations like `fs addmirror`, `fs growto primary`, `fs shrinkto primary`, and so on, will be applied to the primary tier.

Veritas Access includes a feature, cloud as a tier, that lets you move files based on user-specified criteria to cloud storage. This feature is supported for scale-out file systems only. Based on the criteria specified for data movement using the storage tier move operation, data is moved to and from the cloud automatically.

The storage `tier` commands manage file system secondary tiers. All tier commands take a file system name as an argument and perform operations on the secondary tier of that file system.

A file system cannot have more than one tier of the same type.

The `tier` command can be used to perform the following:

- adding/removing/modifying the secondary tier
- adding/removing the cloud tier for scale-out file systems
- adding/removing an S3-compatible or Amazon Glacier cloud tier
- moving files between the primary tier and the cloud tier for scale-out file systems
- listing storage tiers configured for scale-out file systems
- setting policies
- scheduling policies
- locating tier locations of files
- listing files that are located on the primary or the secondary tier
- listing the tier type as S3 or Amazon Glacier
- moving files from the secondary tier to the primary tier

Policy:

Each tier can be assigned a policy. A policy assigned to a file system has three parts:

1. **`File Creation`** This specifies on which tier new files get created.
2. **`Inactive files`** This indicates when a file has to be moved from the primary tier to the secondary tier and can be determined by observing the files' last modification date instead of the last accessed date. For example, if the *days* of a tier is set to 10 days, and if file1 has not been modified for the last 10 days, while file2 has been modified in between, then only file1 will be moved to the secondary tier, whereas file2 still remains in the primary tier.

3. Access Temperature Access temperature is the measure of the number of I/O requests to the file during the period designated by the *period*. In other words, access temperature is the number of read or write requests made to a file over a specified number of 24-hour periods divided by the number of periods. So if the access temperature of a file exceeds more than *minacctemp* (where the access temperature is calculated over a period specified by *period*), then this file is moved from the secondary tier to the primary tier.

Schedule:

When a file is not accessed for more than `policy` number of days, then it is not moved automatically from one tier to another tier. This movement has to be scheduled manually. This schedule has to be set for each file system.

For example, if the user sets the schedule as "0 0 2 * *", and the policy as 60, then on the 2nd of every month, all the files that have not been accessed for more than 60 days are moved from the primary tier to the secondary tier.

The following are the parameters that you can provide while setting a schedule for moving a file from the primary to the secondary tier:

1. File system name: File system whose files are to be moved from the primary to a secondary tier.
2. Minute: This field may contain either an asterisk '*', which implies every minute or a numeric value between 0-59.
3. Hour: This field may contain either an asterisk '*', which implies every hour, or a numeric value between 0-23.
4. Day of month: This field may contain either an asterisk '*', which implies every day of the month, or a numeric value between 1-31.
5. Month: This field may contain either an asterisk '*', which implies run every month, or a numeric value between 1-12. In addition to the numeric values, this field can also accept names of the month as an argument, with the first three letters of the month (all in lowercase) serving as input for the given parameter.
6. Day of Week: This field may contain either an asterisk '*', which implies every day of the week, or a numeric value between 0-6, with 0 being interpreted as Sunday, 1 as Monday, and so on. In addition, this parameter can also accept names, with the first three letters of the month (all in lowercase) serving as input values.
7. Node Name: Cluster node name that executes the schedule. Default node name is the master node.

Note:

1. The parameters minute, hour, day-of-month, month, and day-of-week for `tier schedule modify` can accept numeric values in the form of ranges, where a range is defined as two numbers separated by a hyphen. So if you want to run the tiering schedule between 1 am and 4 am, then you can specify a value 1-4 for `hour` parameter, with the range being inclusive.

2. Similarly, you can specify a step value for other parameters, that is, day-of-month, month, and day-of-week as well. Step values are also allowed after an asterisk '*', so if you want to run a schedule every two hours, you just have to specify '*/2' as the input value for the 'hour' parameter.

So to move files from the primary to the secondary tier every two hours, use the following command:

```
tier schedule modify fs1 * \*/2 * * *
```

17.16.3 OPTIONS

size Size of the tier of the file system (for example, 10m, 10M, 25g, 100G).

nmirrors Number of mirrors.

ncolumns Number of columns.

protection=disk If the protection is set to `disk`, then mirrors are created on separate disks. The disks may or may not be in the same pool.

protection=pool If the protection is set to `pool`, then mirrors are created in separate pools. If not enough space is available, then the file system creation operation fails.

stripeunit=<kilobytes> Set the stripe width of the tier, where possible values of kilobytes are 128, 256, 512, 1024, and 2048.

`tier add cloud` *fs_name tier_name service_name region*

Add a cloud tier to the specified scale-out file system using service name and region. You can specify if you want to add an S3-compatible or Amazon Glacier cloud storage tier by specifying the appropriate option.

The supported regions of the Amazon S3 service are the following:

- California (us-west-1)
- Central (ca-central-1)
- Frankfurt (eu-central-1)
- Ireland (eu-west-1)
- London(eu-west-2)
- Mumbai (ap-south-1)
- Ohio (us-east-2)
- Oregon (us-west-2)
- Seoul (ap-northeast-2)
- Singapore (ap-southeast-1)
- Sydney (ap-southeast-2)
- SaoPaulo (sa-east-1)
- Tokyo (ap-northeast-1)
- Virginia (us-east-1)

The Amazon Glacier tier does not support the following regions: Singapore or SaoPaulo.

Region is irrelevant to an S3-compatible cloud storage service. If you select an S3-compatible service, select `none` as the region. Any other region specified with an S3-compatible service is ignored. Scale-out file system uses AWS signature version 4 to add authentication information to the requests sent to Amazon S3 and S3-compatible services. If an S3-compatible service cannot support AWS signature verion 4, then it can not be added as a cloud tier to a scale-out file system. **Warning:** When a cloud storage service is used as a cloud tier for a scale-out file system, Veritas Access exclusively owns all the buckets and the objects created by Veritas Access. Any attempt to tamper with these buckets or objects outside of Veritas Access corrupts the files represented by those modified objects.

`tier add simple` *fs_name size pool1* [, *disk1*,...]

Add a simple tier to the specified file system on one of the specified pools/disks.

`tier add mirrored` *fs_name size nmirrors pool1* [, *disk1*,...] [protection=disk|pool]

Add a mirrored tier to the specified file system.

`tier add striped` *fs_name size ncolumns pool1* [, *disk1*,...] [stripeunit=<kilobytes>]

Add a striped tier to the specified file system.

`tier add mirrored-stripe` *fs_name size nmirrors ncolumns pool1* [, *disk1*,...] [protection=disk|pool] [stripeunit=<kilobytes>]

Add a mirrored-stripe tier to the specified file system.

`tier add striped-mirror` *fs_name size nmirrors ncolumns pool1* [, *disk1*,...] [protection=disk|pool] [stripeunit=<kilobytes>]

Add a striped-mirror tier to the specified file system.

`tier addcolumn` *fs_name ncolumns pool1* [, *disk1*,...]

Add a column to the tier file system.

`tier rmcolumn` *fs_name*

Remove a column from the tier file system.

`tier remove` *fs_name* [*tier_name*]

Remove a tier from the file system. All the files on the secondary tier get relocated to the primary tier. Ensure that you remove the policy by running the tier policy remove fs_name before running the tier remove command. When removing a cloud tier, the CLI expects that you specify the tier_name as well. If there is data present on the specified tier, then the tier remove operation fails, and the data is not automatically relocated.

`tier list` *fs_name*

Show the list of storage tiers configured on the specified scale-out file system. Display the tier type as S3 or Amazon Glacier.

`tier move start` *fs_name source_tier destination_tier pattern* {atime|mtime} {atime|mtime}

Start a job in the background that moves files that match the criteria from the source_tier to the destination_tier. Pattern identifies the files or directories to move. You can further restrict the files or directories to move by specifying the last accessed time (atime) or the last modified time (mtime). The atime and mtime criteria are optional. The move job returns a job ID for further tracking and control of the job. For the status of the job, use the `tier move list` command.

`tier move dryrun` *fs_name source_tier destination_tier pattern* {atime|mtime} {atime|mtime}

Start a job in the background that does not move the files, but performs a dry run (test) of the move operation. The move dryrun command calculates the number of files that match the criteria. To see the status of the job and to see the number of files calculated, use the tier move list command.

`tier move list`

List the background jobs for move and dry run. This command lists the job IDs and their status.

`tier move status` *JOBID*

This shows the detailed status of the job, which includes the job run type (dryrun or normal), state of the job, the total data to be moved, the total number of files to be moved, data moved so far, the number of files moved so far, and the last file visited.

`tier move abort` *JOBID*

Abort a job in its current state.

`tier addmirror` *fs_name pool1* [, *disk1*,...] [protection=disk|pool]

Add a mirror to the tier of the file system.

`tier rmmirror` *fs_name* [*pool_or_disk_name*]

Remove a mirror from the tier of the file system that is spanning on the specified pools/disks.

`tier listfiles` *fs_name* {primary|secondary}

Show all the files that are on the specified tier.

`tier mapfile` *fs_name file_path*

Show the tier location of the specified file. The path of the file is relative to the file system. For example, `tier mapfile fs1 /a.txt` shows the location of `a.txt`, which is in the root directory of the file system `fs1`.

`tier policy list`

Show the policy for each tiered file system.

`tier policy prune list`

Show the prune policy for each tiered file system.

`tier policy prune modify` *fs_name delete_after*

Change the prune policy of the tiered file system. *delete_after* specifies the number of days after which inactive files are deleted from the secondary tier.

`tier policy modify` *fs_name* {primary|secondary} *days minacctemp period*

Change the policy of the tiered file system. (primary|secondary) specifies the name of the tier on which the new files get created. *days* specifies the number of days after which inactive files are moved from the primary to the secondary tier. *period* is used for calculating the access temperature of a file. If the access temperature of a file exceeds more than *minacctemp*, then this file is moved from the secondary tier to the primary tier.

`tier relocate` *fs_name dirPath*

Relocate directory or file from the secondary tier to the primary tier. This does not relocate NDS (Named Data Stream also includes extended attributes) to the primary tier.

`tier policy remove` *fs_name*

Remove the policy of the tiered file system.

`tier policy prune remove` *fs_name*

Remove the prune policy of the tiered file system.

`tier policy run` *fs_name*

Run the policy of the tiered file system.

`tier schedule list`

Show schedules of all the tiered file systems.

`tier schedule modify` *fs_name minute hour day_of_the_month month year day_of_the_week node_name*

Modify the schedule of the tiered file system.

`tier schedule remove` *fs_name*

Remove the schedule of the tiered file system.

`tier query` *fs_name*

Show the list of files that are moved by running the policy.

`tier allowmetadata yes` *fs_name*

This allows the metadata information on the file system to be written on the secondary tier as well. By default, the secondary tier is not configured for storing the metadata information of the file system. Tiers configured with this option show `metaOK` in the column `SECONDARY TIER` of the `fs list` output.

`tier allowmetadata no` *fs_name*

This restricts the metadata information to the primary tier only. If the primary tier gets full, the writes to the secondary tier are not served, as the metadata updates are restricted to the primary tier only.

`tier setfastresync` *fs_name pool1* [, *disk1*,...]

Set fastresync for the tiers of the file system.

`tier unsetfastresync` *fs_name*

Unset fastresync for the tiers of the file system.

`tier stats {show|reset|usage}` *fs_name tier_name*

Show the access statistics, reset the access statistics to zero, or show the total usage of the specified cloud tier in the `largefs` file system.

`tier stats monitor` *fs_name tier_name* [*interval*]

Monitor the access statistics of the specified cloud tier in the largefs file system. The default interval is five seconds.

17.16.4 EXAMPLES

Create a mirrored file system with name `fs1` with disks from the pools `pool1` and `pool2`. Add a mirrored tier to this file system.

```
Storage> fs create mirrored fs1 100M 2 pool1,pool2
100% [#] Creating mirrored filesystem
Storage> tier add mirrored fs1 100M 2 pool3,pool4
100% [#] Creating mirrored secondary tier of filesystem
```

Add a mirror to the secondary tier of the file system `fs1`.

```
Storage> tier addmirror fs1 pool5
100% [#] Adding mirror to secondary tier of filesystem
```

Add a column to the tier file system.

```
Storage>  tier addcolumn fs1 1 pool1
ACCESS fs INFO V-288-2644 This operation may take some time to complete. You can␣
↪check the status in storage fs list fsname
```

Remove a column from the tier file system.

```
Storage> tier rmcolumn fs1
ACCESS fs INFO V-288-2711 Only one column will be removed.
ACCESS fs INFO V-288-2644 This operation may take some time to complete. You can␣
↪check the status in storage fs list fsname
```

Show the list of files on the primary tier of file system `fs1`.

```
Storage> tier listfiles fs1 primary
/.placement_policy.xml
/lost+found/changelog
```

Show the list of files on the secondary tier of file system `fs1`.

```
Storage> tier listfiles fs1 secondary
```

Show tier location of file a.txt, which is in the root directory of file system `fs1`.

```
Storage> tier mapfile fs1 /a.txt
Tier        Extent Type   File Offset   Extent Size
====        ===========   ===========   ===========
Primary     Data          0 Bytes       1.00 KB
```

The following command sets the policy of file system fs1 such that the new files get created on the primary tier. If a file has not been accessed for more than 7 days, then it is moved from the primary to the secondary tier, and if the access temperature of files in the secondary tier is more than 5, then these files are moved from the secondary tier to the primary tier. Access temperature is calculated over a 3-day period.

```
Storage> tier policy modify fs1 primary 7 5 3
ACCESS fs SUCCESS V-288-0 Successfully modified tiering policy for file system ``fs1``
```

List the tiering policy for all the file systems.

```
Storage> tier policy list
FS                          Create on   Days  MinAccess Temp  PERIOD
========================== =========   ====  ==============  ======
fs1                         primary       7          5          3
```

Modify the prune policy.

```
Storage> tier policy prune modify fs1 180
ACCESS fs SUCCESS V-288-0 Successfully modified the prune policy for file system fs1
```

List the tiering prune policy for all the file systems.

```
Storage> tier policy prune modify fs1 180
FS                        Delete After
========================= =========
fs1                       180
```

Run the tiering policy for file system fs1.

```
Storage> tier policy run fs1
ACCESS fs SUCCESS V-288-0 Successfully ran tiering policy for File system fs1
```

Show the list of files that are moved and or deleted by running the policy.

```
Storage> tier query fs1
/a.txt
/b.txt
/c.txt
/d.txt
```

Set the schedule of file system fs1. Default node name is the master node.

```
Storage> tier schedule modify fs1 1 1 1 * *
ACCESS fs SUCCESS V-288-0 Command 'tier schedule modify' executed successfully for fs1
```

Get the tiering schedule for all the file systems.

```
Storage> tier schedule list
FS                          Minute     Hour      Day    Month  WeekDay   NodeName
==========================  ======     ====      ===    =====  =======   =========
fs1                              1        1        1        *        *    master node
```

Remove the tiering schedule for file system `fs1`.

```
Storage> tier schedule remove fs1
ACCESS fs SUCCESS V-288-0 Command tier schedule remove executed successfully for fs1
```

Remove the policy of file system `fs1`.

```
Storage> tier policy remove fs1
ACCESS fs SUCCESS V-288-0 Successfully removed tiering policy for File system fs1
```

Remove the prune policy of file system `fs1`.

```
Storage> tier policy prune remove fs1
ACCESS fs SUCCESS V-288-0 Successfully removed the Prune policy for File system fs1
```

Remove a mirror from the secondary tier of file system `fs1`.

```
Storage> tier rmmirror fs1
```

Remove the cloudtier from file system `fs1`.

```
Storage> tier remove fs1 cloudtier
```

Allow storing metadata information to the secondary tier.

```
Storage> tier allowmetadata yes fs1
ACCESS fs SUCCESS V-288-0 Configured the secondary tier for storing metadata␣
↪information.
```

Restrict storing metadata information to the primary tier.

```
Storage> tier allowmetadata no fs1
ACCESS fs SUCCESS V-288-0 Configured the secondary tier for storing no metadata␣
↪information.
```

Set fastresync for the tiers of the file system.

```
Storage> tier setfastresync adistr pool1
```

Unset fastresync for the tiers of the file system.

```
Storage> tier unsetfastresync adistr
ACCESS fs SUCCESS V-288-1817 Disabled fast resync on the file system adistr
```

Add the cloud storage of the Amazon S3 service, named `awstest`, in the region of us-west-2 as the cloud tier cloudtier for the scale-out file system `largefs1`.

```
Storage> tier add cloud largefs1 cloudtier awstest us-west-2
ACCESS tier SUCCESS V-288-0 Tier cloudtier is added for largefs1
```

Add S3-compatible cloud storage, named `s3comptest`, as the cloud tier for the scale-out file system `largefs1`. Use `none` as the region for S3-compatible cloud storage.

```
Storage> tier add cloud largefs1 va01 s3comptest none
ACCESS tier SUCCESS V-288-0 Tier va01 is added for largefs1
```

Add an Amazon Glacier cloud storage tier.

```
Storage> tier add cloud gfs2 gtier2 gserv California Glacier
ACCESS tier SUCCESS V-288-0 Tier gtier2 is added for gfs2
Storage> tier list gfs2
TIER NAME            TIER TYPE
=========            =========
primary              DISK
gtier2               CLOUD (Glacier)
```

Show the list of storage tiers for the scale-out file system `largefs1`.

```
Storage> tier list largefs1
TIER NAME            TIER TYPE    PROVIDER      SERVICE NAME
==================== ==========   ========      =============
primary              DISK
cloudtier            CLOUD        AWS           amazon_service
```

Remove the cloud tier cloudtier from the scale-out file system `largefs1`.

```
Storage> tier remove largefs1 cloudtier
ACCESS tier SUCCESS V-288-0 Tier cloudtier is removed
```

Remove the cloud tier va01 from the scale-out file system `largefs1`.

```
Storage> tier remove largefs1 va01
ACCESS tier SUCCESS V-288-0 Tier va01 is removed
```

Move the files with the name pattern *.txt, that have not been accessed within the past 100 days from the disk tier `primary` to the cloud tier `cloudtier`.

```
Storage> tier move start largefs1 primary cloudtier \*.txt atime >100D
ACCESS Tier SUCCESS V-288-0 The job has been started. ID is 1473682883.
```

Move the files with name pattern *.txt, that have been accessed recently in the past 30 days from the cloud tier `cloudtier` to the disk tier `primary`.

```
Storage> tier move start largefs1 cloudtier primary atime <30D
```

Move the files that have not been modified within the past 100 days from the disk tier `primary` to the cloud tier `cloudtier`.

```
Storage> tier move start largefs1 primary cloudtier * mtime >100D
```

Move the files whose name matches the pattern "*.pdf" from tier `primary` to the cloud tier `cloudtier`.

```
Storage> tier move start largefs1 primary cloudtier *.pdf
```

Specify the retrieval options for retrieving data from Amazon Glacier. Expedited retrievals typically complete within 1-5 minutes. Standard retrievals typically complete within 3-5 hours. Bulk retrievals typically complete within 5-12 hours. The Bulk option is the default. An EIO error is returned for read, write, or truncate until the files are moved to the primary tier (which could be 1-5 minutes, or 3-5 hours, or 5-12 hours)

```
Storage> tier move start gfs2 gtier2 primary retrieval_option=Bulk *.txt.
```

Move all the files to the tier `primary`.

```
Storage> tier move start largefs1 cloudtier primary *
```

Perform a dry run of the tier move operation. A dry run can be used for estimating the scope of the tier movement of files.

```
Storage> tier move dryrun largefs1 primary cloudtier *log
ACCESS Tier SUCCESS V-288-0 The job has been started. ID is 1473682883
```

List all the tier move jobs.

```
Storage> tier move list
Job         Fs name   Source Tier   Destination Tier   Pattern         Atime   Mtime
↪ State
==========  ========  ============  =================  ===========     ======  ======
↪ ========
1473684478  largefs1  cloudtier     primary            /vx/largefs1/*  >120s   -
↪ not running
1473684602  largefs1  cloudtier     primary            /vx/largefs1/*  -       -
↪ scanning
```

Show the status of a tier move job.

```
Storage> tier move status 1473682883
Job run type:        normal
Job Status:          running
Total Data (Files):  4.0 G (100)
Moved Data (Files):  100.0 M (10)
Last file visited:   /vx/fstfs/10.txt
```

Show the access statistics of the cloud tier `cloudtier`.

```
Storage> tier stats show largefs1 cloudtier
GET       168
GET bytes 174.5MB
PUT       918
PUT bytes 10.3GB
DELETE    20
```

Monitor the access statistics of the cloud tier `cloudtier` every 10 seconds.

```
Storage> tier stats monitor largefs1 cloudtier 10
        GET     GET bytes       PUT     PUT bytes       DELETE
        0       0               0       0               0
        1       1.4MB           0       0               0
        0       0               3       714.0KB         0
        0       0               0       0               0
```

Reset the access statistics of the cloud tier `cloudtier` to zero. After executing the tier stats reset command, the output for the `tier stats show` command is reset to zero.

```
Storage> tier stats reset largefs1 cloudtier
```

Show the usage of the cloud tier `cloudtier`.

```
Storage> tier stats usage largefs1 cloudtier
Storage Utilized     223.1GB
Number of objects    488
Number of files      231
```

17.16.5 SEE ALSO

disk(1), hba(1), fencing(1), fs(1), pool(1), scanbus(1), snapshot(1), iscsi(1), quota(1)

17.17 uncompress

17.17.1 SYNOPSIS

`uncompress file` *fs_name* fI{file|dir} *res_level*

17.17.2 DESCRIPTION

`uncompress file` ***fs_name*** **fI{file|dir}** ***res_level*** Uncompress a file on the *fs_name* file system.

17.17.3 EXAMPLES

Uncompress file data1.dbf in file system tpcc_arch1 using `high` resource level.

```
Storage> uncompress file tpcc_arch1 data1.dbf high
ACCESS uncompression Success V-288-999 Uncompression completed
```

17.17.4 SEE ALSO

disk(1), hba(1), fencing(1), fs(1), pool(1), scanbus(1), snapshot(1), tier(1), iscsi(1), cifs(1)

18

Support Commands

18.1 support

18.1.1 SYNOPSIS

`debuginfo setlog` *loglevel*

`debuginfo upload` *nodename debug-URL module*

`iostat cpu` [*nodename*] [*interval*] [*count*]

`iostat device` [*nodename*] [*dataunit*] [*interval*] [*count*]

`tethereal export` *url* [*nodename*] [*interface*] [*count*] [*source*]

`tethereal show` [*nodename*] [*interface*] [*count*] [*source*]

`top` [*nodename*] [*iterations*] [*delay*]

`traceroute` *destination* [*source*] [*maxttl*]

`metasave` [*fsname*] [*output_location*]

`services autofix`

`services online` *servicename*

`services show`

`services showall`

18.1.2 DESCRIPTION

These support utility commands are used to debug any problems that occur within the cluster.

18.1.3 OPTIONS

`debuginfo setlog` ***loglevel*** This command set CIFS log level in the cluster.

debuginfo upload ***nodename debug-URL module*** This command uploads debug information of given module from specified node to the external server. The *debug-URL* can be a remote file or directory. If *debug-URL* specifies a remote file, the debuginfo file will be saved by that name; if *debug-URL* specifies a remote directory, the debuginfo file will have the name like nas_debuginfo_nodename_modulename_timestamp.tar.gz. Supported *module* values are [all|generic|cifs|nas|supportconfig].

iostat cpu [*nodename*] [*interval*] [*count*] This command generates CPU and IO statistics information. When the `iostat cpu` command is executed for the first time it contains information since the system was booted. Each subsequent CPU utilization report shows the details since the last report. *nodename* contains the name of the node from where report is generated. By default it accepts 'console' as the Management Console. Interval and count specify the amount of time in seconds between each report and number of reports, respectively.

iostat device [*nodename*] [*dataunit*] [*interval*] [*count*] This command generates the device utilization report. When the `iostat device` command is executed for the first time it contains information since the system was booted. Each subsequent device utilization report shows the details since the last report. It has two optional parameters. The first parameter, *nodename*, contains the name of the node that the report is generated from. By default it uses the term 'console' as the Management Console. The second parameter is the report that can be generated in block(s) or kilobyte(s). By default all of the data is in block(s). Interval and count specify the amount of time in seconds between each report and number of reports, respectively.

tethereal export ***url*** **[*nodename*] [*interface*] [*count*] [*source*]** This command exports the network traffic details. The *url* provides the location to export the network traffic details. The default file name is `tethereal.log`, if it is not specified in *url*. The *nodename* contains the name of the node from where traffic details are generated. By default it accepts 'console' as the Management Console. The *interface* specifies the network interface for packet capture. The *count* specifies the maximum number of packets to read. A packet can be filtered by specifying the *source* node.

tethereal show [*nodename*] [*interface*] [*count*] [*source*] This command displays the network traffic details for a given *nodename*. By default it accepts 'console' as the Management Console. The *interface* specifies the network interface for the packet capture. The *count* specifies the maximum number of packets to read. A packet can be filtered by specifying the *source* node.

top [*nodename*] [*iterations*] [*delay*] This command provides the dynamic look of processor activity in real time and displays the list of CPU intensive tasks. It shows resources consumed by users and processes at a given time for a given *nodename*. The *iterations* parameter specifies the number of iterations (the default is 3). The *delay* parameter specifies the delay between screen updates (the default is 5 seconds).

traceroute ***destination*** **[*source*] [*maxttl*]** This command shows the route between two nodes by orderly listing all of the intermediate nodes over the network. It displays how a node is connected to the internet and eventually to a specified target *destination*; or in between the gateway that is discarding packets. The *source* and *maxttl* specify the source node name to begin the trace and the maximum number of hops (the default is 7).

metasave [*fsname*] [*output_location*] Collects Metasave image of a regular or a largefs File System specified by *fsname*. The Metasave image is stored at the directory location specified by *output_location*. For regular File System a single Metasave image will be created under *output_location*. For a largefs File System multiple Metasave images will be produced depending on number of container File Systems inside the largefs File System. For largefs File System the namespace mapping database is also dumped.

services autofix Attempts to fix any faults with any services on all of the running nodes in the cluster.

services online ***servicename*** Brings a service online. If *servicename* is a parallel service that can

be brought online on all nodes, an attempt is made to bring the service online on all of the nodes. If *servicename* is a failover service, an attempt is made to bring the service online on any of the running nodes in the cluster. If the *servicename* is already online, no action is taken.

services show Displays the state of the services on all the running nodes in the cluster. State of the IPs and file systems are shown only if they are not online. It might also attempt to fix any faults with any services.

services showall Displays the state of all of the services on all the running nodes in the cluster. It might also attempt to fix any faults with any of the services.

18.1.4 EXAMPLES

Set CIFS loglevel to 10 in the cluster.

```
Support> debuginfo setlog 10
```

Upload all debug information to an FTP server.

```
Support> debuginfo upload node1_1
ftp://admin@ftp.docserver.veritas.com/patches/ all
```

Upload CIFS related debug information to an SCP server.

```
Support> debuginfo upload node1_1
scp://root@server.veritas.com:/tmp/ cifs
```

Shows the cpu statistics.

```
Support> iostat cpu test_01 1 1
Linux 2.6.16.60-0.21-smp (test_01)         Wednesday 08 July 2009

avg-cpu:  %user   %nice    %sys %iowait   %idle
           0.14    0.01    0.12    0.01   99.72
```

Shows the device statistics.

```
Support> iostat device test_01 Blk 1 1
Linux 2.6.16.60-0.21-smp (test_01)         Wednesday 08 July 2009

Device:            tps   Blk_read/s   Blk_wrtn/s   Blk_read   Blk_wrtn
sda               2.91         6.54        63.73     628792    6123480
sdb               0.00         0.01         0.00        904          0
sdc               0.00         0.03         0.00       2514          0
sdd               0.02         0.69         0.06      66034       5409
sde               0.01         0.62         0.59      59929      56314
sdf               0.00         0.03         0.00       2514          0
```

Show the network traffic.

```
Support> tethereal show test_01 pubeth0 5 172.31.168.140
  0.000000 172.31.168.140 -> 10.209.105.147 ICMP Echo (ping) request
  0.000276 10.209.105.147 -> 172.31.168.140 ICMP Echo (ping) reply
  0.000473 10.209.105.147 -> 172.31.168.140 SSH Encrypted response packet len=112
  0.000492 10.209.105.147 -> 172.31.168.140 SSH Encrypted response packet len=112
  0.000845 172.31.168.140 -> 10.209.105.147 TCP 50777 > 22 [ACK] Seq=0 Ack=112␣
↪Win=3624 Len=0
```

Export the network traffic.

```
Support> tethereal export scp://user1@172.31.168.140:~/
Password: *******
Capturing on pubeth0 ...
Uploading network traffic details to scp://user1@172.31.168.140:~/ is completed.
```

Display the dynamic real-time view of the tasks.

```
Support> top test_01 1 1
top - 16:28:27 up 1 day,  3:32,  4 users,  load average: 1.00, 1.00, 1.00
Tasks: 336 total,   1 running, 335 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.1% us,  0.1% sy,  0.0% ni, 99.7% id,  0.0% wa,  0.0% hi,  0.0% si
Mem:  16405964k total,  1110288k used, 15295676k free,   183908k buffers
Swap:  1052248k total,        0k used,  1052248k free,   344468k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
6314 root      15   0  5340 1296  792 R  3.9  0.0   0:00.02 top
    1 root      16   0   640  260  216 S  0.0  0.0   0:04.86 init
```

Trace the route to network host.

```
Support> traceroute www.veritas.com test_01 10
traceroute to www.veritas.com (8.14.104.56), 10 hops max, 40 byte packets
1  10.209.104.2  0.337 ms   0.263 ms   0.252 ms
2  10.209.186.14  0.370 ms   0.340 ms   0.326 ms
3  puna-spi-core-b02-vlan105hsrp.net.veritas.com (143.127.185.130)  0.713 ms   0.525␣
↪ms   0.533 ms
4  143.127.185.197  0.712 ms   0.550 ms   0.564 ms
5  10.212.252.50  0.696 ms   0.600 ms   78.719 ms
```

Display the state of important services.

```
Support> services show
Verifying cluster state..........done

                       test
Service            1        2
-------         -------- --------
nfs               ONLINE STARTING
cifs              ONLINE   ONLINE
ftp               ONLINE   ONLINE
http              ONLINE   ONLINE
backup            ONLINE  OFFLINE
console           ONLINE  OFFLINE
nic_pubeth0       ONLINE   ONLINE
nic_pubeth1       ONLINE   ONLINE
fs_manager        ONLINE   ONLINE
10.216.50.132    FAULTED  FAULTED
```

Display the state of all of the services.

```
Support> services showall

                       test
Service            1        2
-------         -------- --------
nfs               ONLINE STARTING
cifs              ONLINE   ONLINE
```

```
ftp                ONLINE   ONLINE
http               ONLINE   ONLINE
backup             ONLINE  OFFLINE
console            ONLINE  OFFLINE
nic_pubeth0        ONLINE   ONLINE
nic_pubeth1        ONLINE   ONLINE
fs_manager         ONLINE   ONLINE
10.216.115.199    OFFLINE   ONLINE
10.216.115.200     ONLINE  OFFLINE
10.216.115.201    OFFLINE   ONLINE
10.216.115.202     ONLINE  OFFLINE
10.216.50.132     FAULTED  FAULTED
/vx/fs1            ONLINE   ONLINE
```

Bring a service online.

```
Support> services online 10.216.50.132
Support> services show

                       test
Service             1        2
-------          -------- --------
nfs                ONLINE STARTING
cifs               ONLINE   ONLINE
ftp                ONLINE   ONLINE
http               ONLINE   ONLINE
backup             ONLINE  OFFLINE
console            ONLINE  OFFLINE
nic_pubeth0        ONLINE   ONLINE
nic_pubeth1        ONLINE   ONLINE
fs_manager         ONLINE   ONLINE
10.216.50.132     OFFLINE STARTING
```

Autofix all the services.

```
Support> services show

                       test
Service             1        2
-------          -------- --------
nfs                ONLINE STARTING
cifs               ONLINE   ONLINE
ftp                ONLINE   ONLINE
http               ONLINE   ONLINE
backup             ONLINE  OFFLINE
console            ONLINE  OFFLINE
nic_pubeth0        ONLINE   ONLINE
nic_pubeth1        ONLINE   ONLINE
fs_manager         ONLINE   ONLINE
10.216.50.132     FAULTED  FAULTED

Support> services autofix
Attempting to fix service faults..........done
Support> services show

                       test
Service             1        2
-------          -------- --------
```

```
nfs                ONLINE STARTING
cifs               ONLINE   ONLINE
ftp                ONLINE   ONLINE
http               ONLINE   ONLINE
backup             ONLINE  OFFLINE
console            ONLINE  OFFLINE
nic_pubeth0        ONLINE   ONLINE
nic_pubeth1        ONLINE   ONLINE
fs_manager         ONLINE   ONLINE
10.216.50.132     OFFLINE STARTING
```

18.2 debuginfo

18.2.1 SYNOPSIS

`debuginfo setlog cifs` *loglevel*

`debuginfo setlog nbu database` *loglevel*

`debuginfo setlog nbu global` *loglevel*

`debuginfo setlog nbu enable critical`

`debuginfo setlog nbu enable robust`

`debuginfo upload` *nodename debug-URL module*

18.2.2 DESCRIPTION

The `debuginfo` commands can set the CIFS or the NBU client log levels in the cluster, as well as upload debug information to an external FTP or SCP server.

18.2.3 OPTIONS

`debuginfo setlog cifs` ***loglevel*** Set CIFS log level in the cluster. Valid log level value ranges from 0 to 10, 10 being the most detailed. Default is 2.

`debuginfo setlog nbu database` ***loglevel*** Set NetBackup database debugging log level in the cluster. Valid log level value ranges from 1 to 5, 5 being the most detailed.

`debuginfo setlog nbu global` ***loglevel*** Set the NetBackup global debugging log level in the cluster. Valid log level value ranges from 1 to 5, 5 being the most detailed.

`debuginfo setlog nbu enable robust` Enable the NetBackup client to perform robust logging in the cluster.

`debuginfo setlog nbu enable critical` Enable the NetBackup client to perform critical process logging in the cluster.

`debuginfo upload` ***nodename debug-URL module*** Upload debugging information of given module from specified node to the external server. The *debug-URL* can be a remote file or directory. If *debug-URL* specifies a remote file, the debuginfo file will be saved by that name; if *debug-URL* specifies a remote directory, the debuginfo file uses a name like **nas_debuginfo_nodename_modulename_timestamp.tar.gz**. Supported *module* values are [`all|generic|cifs|nas|netbackup|supportconfig`]. Use `all` to collect all information needed for debug, `generic` to collect all information except Veritas product information, `cifs` to collect CIFS related information, `nas` to collect product-related information, `netbackup` to collect NetBackup-related information, `supportconfig` to collect system support-related information.

18.2.4 EXAMPLES

Set CIFS log level to 10 in the cluster.

```
Support> debuginfo setlog cifs 10
```

Upload all debug information to an FTP server.

```
Support> debuginfo upload node1_1
ftp://admin@ftp.docserver.veritas.com/patches/ all
```

Upload CIFS related debug information to an SCP server.

```
Support> debuginfo upload node1_1
scp://root@server.veritas.com:/tmp/ cifs
```

Upload CIFS related debug information to file in debug_dir directory on a node.

```
Support> debuginfo upload node1_1
file:///debug_dir/ cifs
```

Set NetBackup client database log level to 3 in the cluster.

```
Support> debuginfo setlog nbu database 3
```

18.3 iostat

18.3.1 SYNOPSIS

`iostat cpu` [*nodename*] [*interval*] [*count*]

`iostat device` [*nodename*] [*dataunit*] [*interval*] [*count*]

18.3.2 DESCRIPTION

`iostat` commands display the CPU and I/O statistics.

18.3.3 OPTIONS

`iostat cpu` [*nodename*] [*interval*] [*count*] This command generates CPU and I/O statistics. When the `iostat cpu` command is executed for the first time, the output contains information from when the system was booted. Each subsequent CPU utilization report displays the details since the last report. The *nodename* contains the name of the node that the report is generated from. By default, it uses the term 'console' for the Management Console. Interval and count specify the amount of time in seconds between each report and number of reports, respectively.

`iostat device` [*nodename*] [*dataunit*] [*interval*] [*count*] This command generates the utilization report of the device. When the `iostat device` command is executed for the first time, it contains information from when the system was booted. Each subsequent device utilization report shows the details since the last report. It has two optional arguments. The first argument,*nodename*, contains the name of the node that the report is generated from. By default, it uses the term 'console' for the Management Console. The report can be generated in terms of block(s) or kilobyte(s) by specifying the second argument. Unless specified, all of the data will be in block(s). Interval and count specify the amount of time in seconds between each report and number of reports, respectively.

18.3.4 EXAMPLES

Shows the cpu statistics.

```
Support> iostat cpu test_01 1 1
Linux 2.6.16.60-0.21-smp (test_01)         Wednesday 08 July 2009

avg-cpu:  %user   %nice    %sys %iowait   %idle
           0.14    0.01    0.12    0.01   99.72
```

Shows the device statistics.

```
Support> iostat device test_01 Blk 1 1
Linux 2.6.16.60-0.21-smp (test_01)         Wednesday 08 July 2009

Device:            tps   Blk_read/s   Blk_wrtn/s   Blk_read   Blk_wrtn
sda               2.91         6.54        63.73     628792    6123480
sdb               0.00         0.01         0.00        904          0
sdc               0.00         0.03         0.00       2514          0
sdd               0.02         0.69         0.06      66034       5409
sde               0.01         0.62         0.59      59929      56314
sdf               0.00         0.03         0.00       2514          0
```

18.3.5 SEE ALSO

tethereal(1), top(1), traceroute(1)

18.4 metasave

18.4.1 SYNOPSIS

`metasave` [*fsname*] [*output_location*]

18.4.2 DESCRIPTION

`metasave` command collects metasave image of a regular or scale-out file system for debugging purposes.

18.4.3 OPTIONS

`metasave` **[*fsname*] [*output_location*]** Collects Metasave image of a regular or a scale-out file system specified by *fsname*. The Metasave image is stored at the directory location specified by *output_location*. For a CFS file system, a single Metasave image is created under *output_location*. For a scale-out file system, multiple Metasave images are produced depending on the number of container file systems inside the scale-out file system. For the scale-out file system, the namespace mapping database is also dumped.

18.4.4 EXAMPLES

Collect metasave of file system *testfs* and store it under **/tmp/meta_out_dir**.

```
Support> metasave testfs /tmp/meta_out_dir
Collecting Metasave image of File System testfs. This may take some time...
SUCCESS: Metasave image of testfs collected succesfully. TAR Image is stored at /tmp/
↪meta_out_dir/metasave_tempfs.tar.
```

18.4.5 NOTE

File system must be offline on all the cluster nodes to create a consistent metasave image. Bring the file system offline before collecting metasave using the `Storage>fs offline` command. Metasave image collection is a time-consuming operation. Total time taken depends on the amount of metadata information present in the file system. Time taken for scale-out file system can be significantly higher. Other Veritas Access operations can be run from a separate terminal while the metasave collection is in progress.

18.4.6 SEE ALSO

iostat(1), debuginfo(1), fs(1)

18.5 services

18.5.1 SYNOPSIS

`services autofix`

`services online` *servicename*

`services show`

`services showall`

18.5.2 OPTIONS

servicename Name of the service that needs to be online.

`services autofix` Attempts to fix any faults with any services on all of the running nodes of the cluster.

`services online` ***servicename*** Brings a service online. If *servicename* is a parallel service that can be onlined on all nodes, an attempt is made to bring the service online on all of the nodes. If *servicename* is a failover service, an attempt is made to bring the service online on any of the running nodes in the cluster. If *servicename* is already online, no action is taken.

`services show` Shows the state of the services on all of the running nodes of the cluster. The state of the IPs and file systems are shown only if they are not online. It might also attempt to fix any faults with any of the services.

`services showall` Shows the state of all the services on all of the running nodes of the cluster. It might also attempt to fix any faults with any of the services.

18.5.3 EXAMPLES

Display the state of important services.

```
Support> services show
Verifying cluster state..........done

                       test
Service            1        2
-------         -------- --------
nfs               ONLINE STARTING
cifs              ONLINE   ONLINE
ftp               ONLINE   ONLINE
http              ONLINE   ONLINE
backup            ONLINE  OFFLINE
console           ONLINE  OFFLINE
nic_pubeth0       ONLINE   ONLINE
nic_pubeth1       ONLINE   ONLINE
fs_manager        ONLINE   ONLINE
10.216.50.132    FAULTED  FAULTED
```

Display the state of all of the services.

```
Support> services showall
```

```
                        test
Service            1        2
-------         -------- --------
nfs               ONLINE STARTING
cifs              ONLINE   ONLINE
ftp               ONLINE   ONLINE
http              ONLINE   ONLINE
backup            ONLINE  OFFLINE
console           ONLINE  OFFLINE
nic_pubeth0       ONLINE   ONLINE
nic_pubeth1       ONLINE   ONLINE
fs_manager        ONLINE   ONLINE
10.216.115.199   OFFLINE   ONLINE
10.216.115.200    ONLINE  OFFLINE
10.216.115.201   OFFLINE   ONLINE
10.216.115.202    ONLINE  OFFLINE
10.216.50.132    FAULTED  FAULTED
/vx/fs1           ONLINE   ONLINE
```

Bring a service online.

```
Support> services online 10.216.50.132
Support> services show

                        test
Service            1        2
-------         -------- --------
nfs               ONLINE STARTING
cifs              ONLINE   ONLINE
ftp               ONLINE   ONLINE
http              ONLINE   ONLINE
backup            ONLINE  OFFLINE
console           ONLINE  OFFLINE
nic_pubeth0       ONLINE   ONLINE
nic_pubeth1       ONLINE   ONLINE
fs_manager        ONLINE   ONLINE
10.216.50.132    OFFLINE STARTING
```

Autofix all of the services.

```
Support> services show

                        test
Service            1        2
-------         -------- --------
nfs               ONLINE STARTING
cifs              ONLINE   ONLINE
ftp               ONLINE   ONLINE
http              ONLINE   ONLINE
backup            ONLINE  OFFLINE
console           ONLINE  OFFLINE
nic_pubeth0       ONLINE   ONLINE
nic_pubeth1       ONLINE   ONLINE
fs_manager        ONLINE   ONLINE
10.216.50.132    FAULTED  FAULTED

Support> services autofix
Attempting to fix service faults..........done
```

```
Support> services show

                       test
Service            1        2
-------         -------- --------
nfs               ONLINE STARTING
cifs              ONLINE   ONLINE
ftp               ONLINE   ONLINE
http              ONLINE   ONLINE
backup            ONLINE  OFFLINE
console           ONLINE  OFFLINE
nic_pubeth0       ONLINE   ONLINE
nic_pubeth1       ONLINE   ONLINE
fs_manager        ONLINE   ONLINE
10.216.50.132    OFFLINE STARTING
```

18.6 tethereal

18.6.1 SYNOPSIS

`tethereal export` *url* [*nodename*] [*interface*] [*count*] [*source*]

`tethereal show` [*nodename*] [*interface*] [*count*] [*source*]

18.6.2 DESCRIPTION

`tethereal` commands dump and analyze the network traffic.

18.6.3 OPTIONS

`tethereal export` *url* [*nodename*] [*interface*] [*count*] [*source*] This command exports the network traffic details. *url* provides the location to export the network traffic details. Default file name is `tethereal.log`, if it is not specified in *url*. *nodename* contains the name of the node from where traffic details are generated; by default, it takes 'console' for the Management Console. *interface* and *count* specify the network interface for packet capture and maximum number of packets to read, respectively. Packet can be filtered by specifying *source* node. Press the `Ctrl+C` to stop the capture process and upload traffic details to the *url* site.

`tethereal show` [*nodename*] [*interface*] [*count*] [*source*] This command displays the network traffic details for a given *nodename*; by default, it takes 'console' for the Management Console. *interface* and *count* specify the network interface for packet capture and maximum number of packets to read, respectively. Packet can be filtered by specifying *source* node.

18.6.4 EXAMPLES

Show the network traffic.

```
Support> tethereal show test_01 pubeth0 5 172.31.168.140
  0.000000 172.31.168.140 -> 10.209.105.147 ICMP Echo (ping) request
  0.000276 10.209.105.147 -> 172.31.168.140 ICMP Echo (ping) reply
  0.000473 10.209.105.147 -> 172.31.168.140 SSH Encrypted response packet len=112
  0.000492 10.209.105.147 -> 172.31.168.140 SSH Encrypted response packet len=112
  0.000845 172.31.168.140 -> 10.209.105.147 TCP 50777 > 22 [ACK] Seq=0 Ack=112␣
↪Win=3624 Len=0
```

Export the network traffic.

```
Support> tethereal export scp://user1@172.31.168.140:~/
Password: *******
Capturing on pubeth0 ...
Uploading network traffic details to scp://user1@172.31.168.140:~/ is completed.
```

18.6.5 SEE ALSO

iostat(1), top(1), traceroute(1)

18.7 top

18.7.1 SYNOPSIS

`top` [*nodename*] [*iterations*] [*delay*]

18.7.2 DESCRIPTION

`top` command provides dynamic look of processor activity in real-time.

18.7.3 OPTIONS

`top` [*nodename*] [*iterations*] [*delay*] The `top` command provides a dynamic look of processor activity in real-time and displays the list of CPU intensive tasks. It shows resources consumed by users and procecesses at a given time for a given *nodename*. *iterations* and *delay* specify the number of iterations (by default 3) and delay between screen updates (by default 5 seconds), respectively.

18.7.4 EXAMPLES

Display the dynamic real-time view of a task.

```
Support> top test_01 1 1
top - 16:28:27 up 1 day,  3:32,  4 users,  load average: 1.00, 1.00, 1.00
Tasks: 336 total,   1 running, 335 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.1% us,  0.1% sy,  0.0% ni, 99.7% id,  0.0% wa,  0.0% hi,  0.0% si
Mem:  16405964k total,  1110288k used, 15295676k free,   183908k buffers
Swap:  1052248k total,        0k used,  1052248k free,   344468k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
6314 root      15   0  5340 1296  792 R  3.9  0.0   0:00.02 top
    1 root      16   0   640  260  216 S  0.0  0.0   0:04.86 init
```

18.7.5 SEE ALSO

iostat(1), tethereal(1), traceroute(1)

18.8 traceroute

18.8.1 SYNOPSIS

`traceroute` *destination* [*source*] [*maxttl*]

18.8.2 DESCRIPTION

`traceroute` command shows the route from a node to a specified target.

18.8.3 OPTIONS

`traceroute` ***destination*** **[*source*] [*maxttl*]** `traceroute` shows the route between two nodes by orderly listing all the intermediate nodes over the network. It displays how a node is connected to the Internet and eventually to a specified target *destination*; or an in-between gateway that is discarding packets. *source* and *maxttl* specify the source node name to begin trace and maximum number of hops (by default 7).

18.8.4 EXAMPLES

Trace the route to the network host.

```
Support> traceroute www.veritas.com test_01 10
traceroute to www.veritas.com (8.14.104.56), 10 hops max, 40 byte packets
1  10.209.104.2  0.337 ms   0.263 ms   0.252 ms
2  10.209.186.14  0.370 ms   0.340 ms   0.326 ms
3  puna-spi-core-b02-vlan105hsrp.net.veritas.com (143.127.185.130)  0.713 ms   0.525␣
↪ms   0.533 ms
4  143.127.185.197  0.712 ms   0.550 ms   0.564 ms
5  10.212.252.50  0.696 ms   0.600 ms   78.719 ms
```

18.8.5 SEE ALSO

iostat(1), tethereal(1), top(1)

19 System Commands

19.1 system

19.1.1 SYNOPSIS

`clock show`

`clock set` *time day month year*

`clock timezone` *timezone*

`clock regions` [*region*]

`config export local` *file_name*

`config export remote` *URL*

`config import local` *file_name*

{ network|admin|all|report|system|cluster_specific|all_except_cluster_specific|
nfs|cifs|ftp|backup|replication|storage_schedules|storage_quota|storage_fs_alert|
compress_schedules|defrag_schedules|storage_dedup|smartio }

`config import remote` *URL*

{ network|admin|all|report|system|cluster_specific|all_except_cluster_specific|
nfs|cifs|ftp|backup|replication|storage_schedules|storage_quota|storage_fs_alert|
compress_schedules|defrag_schedules|storage_dedup|smartio }

`config delete` *file_name*

`config list`

`guienable`

`guidisable`

`guistatus`

`kms config list`

`kms config status`

`kms config server` *server_ip server_port*

`kms config delete`

`kms certificate generate`

`kms certificate import_server_cert`

`kms certificate import_client_key`

`kms certificate import_client_cert`

`license add` *<path_of_key_file>*

`license list`

`license list details`

`more enable`

`more disable`

`more status`

`ntp servername` *server-name*

`ntp show`

`ntp enable`

`ntp disable`

`ntp sync`

`option show nfsd`

`option show cfsmount_ontimeout`

`option show dmpio`

`option show ninodes`

`option show tunefstab` *fs-name*

`option show dmptune`

`option modify nfsd` *number* [*nodename*]

`option modify dmpio` {enclosure *enclr_name* | arrayname

array_name | arraytype *{A/A|A/P|...}* }

iopolicy={adaptive | adaptiveminq | balanced

minimumq | priority | round-robin | singleactive}

[nodename={nodename|"“}]

`option modify ninodes` *{ number | Auto }*

`option modify tunefstab` *initial_extent_size value fs-name*

`option modify tunefstab` *read_nstream value fs-name*

`option modify tunefstab` *read_pref_io value fs-name*

`option modify tunefstab` *write_nstream value fs-name*

`option modify tunefstab` *write_pref_io value fs-name*

`option modify tunefstab` *write_throttle value fs-name*

`option modify dmptune` *dmp_path_age value*

`option modify dmptune` *dmp_health_time value*

`stat sys` [*node*]

`stat cluster`

`stat dmp` [*node*]

`stat fsio` [*fsname*]

`stat all` [*node*]

`stat rdma` [*node*]

`webservices server` [start|stop|status]

`webservices SSL certificate` *certificate_path*

`webservices SSL disable`

`webservices SSL enable`

`webservices user password`

19.1.2 DESCRIPTION

The `system` commands set or show the date and time of the system, and start, stop, sync or check the status of the NTP server. It also contains commands related to showing cluster-wide performance statistics and commands. An administrator can enable or disable the more filter on output of the admin console with the help of the `more` command present in this category. It also contains the `option` command display for configuring the tunable parameters. It also contains the `license` commands which provide options to perform necessary operations related to licensing. It also contains the `kms` commands that register or unregister the cluster with the KMS server.

19.1.3 OPTIONS

`clock show` Display the current time and date of the system.

`clock set` ***time day month year*** Set the current date and time of the system. The format of time, day, month, and year is given below.:

```
*time*  : HH:MM:SS using a 24 hour clock
*day*   : 1..31
*month* : January | February | March | April | May | June | July |␣
↪August | September | October | November | December
*year*  : YYYY
```

`clock timezone` ***timezone*** Set the timezone to the specified name of the time zone.

`clock regions` [***region***] Display the list of regions. Regions can be one of the following:

```
{ Africa, America, Asia, Australia, Canada, Europe, GMT-offset, Pacific,␣
↪US }
```

`config export local` ***file_name*** Save the configuration in a local file.

`config export remote` ***URL*** Save the configuration on a remote machine. Exported File is named export.tar.gz if no file name is specified in the URL. `Note`: When exporting files with ftp URL, the given path is considered relative to the current working directory of the session. For example, to get the file named README from your home directory on your ftp site, use:

```
ftp://user:passwd@my.site.com/README
```

But if you want to get the README file from the root directory of the site, you need to specify the absolute file name:

```
ftp://user:passwd@my.site.com//README
```

(that is, with an extra slash in front of the file name.)

config import local ***file_name***

{ network|admin|all|report|system|cluster_specific|all_except_cluster_specific| nfs|cifs|ftp|backup|replication|storage_schedules| compress_schedules|defrag_schedules|storage_dedup|smartio}

Import the specified configuration saved in a local file. `Note` For importing cifs local users/groups you have to import `admin` module also.

config import remote ***URL***

{ network|admin|all|report|system|cluster_specific|all_except_cluster_specific| nfs|cifs|ftp|backup|replication|storage_schedules| compress_schedules|defrag_schedules|storage_dedup|smartio}

Import the specified configuration from a remote machine. `Note` For importing cifs local users/groups you have to import `admin` module also. `Note`: When exporting files with ftp URL, the given path is considered relative to the current working directory of the session. For example, to get the file named README from your home directory on your ftp site, use:

```
ftp://user:passwd@my.site.com/README
```

But if you want to get the README file from the root directory of the site, you need to specify the absolute file name:

```
ftp://user:passwd@my.site.com//README
```

(i.e. with an extra slash in front of the file name.)

config delete ***file_name*** Delete the locally saved configuration file.

config list View the list of locally saved configuration files.

guienable The system guienable command enables use of the GUI Management Server.

guidisableable The system guidisable command disables use of the GUI Management Server.

guistatus The system guistatus displays status of the GUI Management Server.

kms config list Display the registered KMS server details.

kms config status List the status of SSL certificate files that are configured on the cluster.

kms config server ***server_ip server_port*** Register the KMS server with cluster.

kms config delete Delete the KMS server details.

kms certificate generate Generate SSL self-signed keypair on cluster. It is advised that cluster nodes' time settings are synchronized with NTP server before generating self-signed certificates.

kms certificate import_server_cert Import the KMS servers public key.

kms certificate import_client_key Import the SSL key file of the cluster that is used in CSR.

kms certificate import_client_cert Import the CA signed public key of the cluster.

license add ***<path_of_key_file>*** Adds or installs the license using the .slf key file which is provided. The scp protocol is also supported for adding license. The scp path should be specified: scp://user@ip:<path_of_key_file>

license list Lists the type of license installed on the cluster. The license type can be trialware (trial license for 60 days), subscription (license with validity of 1 year) or perpetual (permanent license).

license list details Lists the details of the license installed on the cluster such as the product name, version, start date, end date, license type, current state etc.

more enable Enable the more filter with the output of the admin console command.

more disable Disable the more filter with the output of the admin console command.

more status Display the status of the more filter.

ntp show Display the NTP status and server name.

ntp enable Enable NTP on all the nodes in the cluster.

ntp disable Disable NTP on all the nodes in the cluster.

ntp servername ***server-name*** Set the NTP server on all the nodes in the cluster. Use *127.127.1.0* as the server name to set the local clock.

ntp sync Sync date with NTP server on all the nodes in the cluster. `Note` NTP should be disabled before running "ntp sync" commmand.

option show nfsd Display the number of nfs daemons for each node in a cluster.

option show cfsmount_ontimeout Display cfsmount online timeout.

option show dmpio Display the type of iopolicy corresponding to enclosure, arrayname, arraytype for each node.

option show ninodes Display the global inode cache size.

option show tunefstab ***fs-name*** Display the file-system specific tunefstab parameters and their value for the specified file system.

option show dmptune Display the value of the dmptune attribute.

option modify nfsd ***number*** **[*****nodename*****]** Modify the number of daemons. The range for the number of daemons is from 1 to 512.

option modify cfsmount_ontimeout ***Value*** Modify cfsmount online timeout. The range is from 300 to 9000. The default timeout value is 1200 seconds.

option modify dmpio {enclosure ***enclr_name*** **| arrayname**

array_name **| arraytype** ***{A/A|A/P|...}*** **}** iopolicy={adaptive | adaptiveminq | balanced minimumq | priority | round-robin | singleactive} [nodename={nodenamel"“}]

Modify the dmpio policy corresponding to enclosure, arrayname, arraytype.

option modify ninodes ***{ number | Auto }*** Modify the global inode cache size.

option modify tunefstab ***write_throttle value fs-name*** Modify the write_throttle parameter value for the specified file system. The parameter write_throttle lets you lower the number of dirty pages per file that the file system generates before writing them to disk. After the number of dirty pages for a file reaches the write_throttle threshold, the file system starts flushing pages to disk even if free memory is still available. The valid range for write_throttle is 0 to 2048 pages. The default value is 0 which impies there is no write_throttle.

option modify tunefstab ***initial_extent_size value fs-name*** Modify the initial_extent_size parameter value for the specified file system. The default value for initial_extent_size is 1 file system block. The valid range for initial_extent_size is 1 to 32K blocks and must be the power of 2.

option modify tunefstab ***read_nstream value fs-name*** Modify the read_nstream parameter value for the specified file system. The valid range for read_nstream parameter is 1 to number of stripe columns in the associated volume. For the media server workload file system, the recommended read_nstream parameter value is 1.

option modify tunefstab ***read_pref_io value fs-name*** Modify the read_pref_io parameter value for the specified file system. The valid values for read_pref_io are 64k, 128k, 256k, or 512k.

option modify tunefstab ***write_nstream value fs-name*** Modify the write_nstream parameter value for the specified file system. The valid range for read_nstream parameter is 1 to number of stripe columns in the associated volume.

option modify tunefstab ***write_pref_io value fs-name*** Modify the write_pref_io parameter value for the specified file system. The valid values for write_pref_io are 64k, 128k, 256k, or 512k.

option modify dmptune ***dmp_path_age value*** Modify the value of dmp_path_age.

option modify dmptune ***dmp_health_time value*** Modify the value of dmp_health_time.

stat sys [***node***] Display system-related statistics.

stat cluster Display cluster-wide statistics.

stat dmp [***node***] Display DMP-related statistics.

stat fsio [***fsname***] Display filesystem IO statistics.

stat all [***node***] Display system and DMP-related statistics at a time.

stat rdma [***node***] Display rdma statistics.

webservices server [start|stop|status] Start, stop, or display the status of the REST server.

webservices SSL certificate ***certificate_path*** Set the certificate file of SSL for the REST server.

webservices SSL disable Disable SSL for the REST server.

webservices SSL enable Enable SSL for the REST server.

webservices user password Set the user password. Currently, Veritas Access supports only one user, the manila user.

19.1.4 SEE ALSO

clock(1), ntp(1), option(1), stat(1), webservices(1), license(1), kms(1)

19.2 clock

19.2.1 SYNOPSIS

`clock show`

`clock set` *time day month year*

`clock timezone` *timezone*

`clock regions` [*region*]

19.2.2 DESCRIPTION

The system `clock` commands set or show the date and time of the system, including the time zones and the list of regions. Changing cluster time is dangerous and may affect replication,DST,snapshot and other functionalities.

19.2.3 OPTIONS

`clock show` Display the current time and date of the system.

`clock set` *time day month year*

Set the current date and time of the system. The format of time, day, month, and year is given below.
time : HH:MM:SS using a 24 hour clock *day* : 1..31 *month* : January | February | March | April | May | June | July | August | September | October | November | December *year* : YYYY

`clock timezone` *timezone* Set the timezone to the specified name of the time zone.

`clock regions` [*region*] Display the list of regions. Regions can be one of the following:

```
{ Africa, America, Asia, Australia, Canada, Europe, GMT-offset, Pacific,
↪US }
```

19.2.4 EXAMPLES

Display the current date and time.

```
System> clock show
Mon Jul 13 07:10:39 EDT 2009
```

Set 1 January 2014 on all the nodes.

```
System> clock set 13:00:00 1 January 2014
```

List all of the timezones in region `Asia`.

```
System> clock regions Asia
```

Set `Calcutta` as the timezone.

```
System> clock timezone Calcutta
```

19.2.5 SEE ALSO

ntp(1)

19.3 config

19.3.1 SYNOPSIS

`config export local` *file_name*

`config export remote` *URL*

`config import local` *file_name [config_type]*

{ network|admin|all|report|system|cluster_specific|all_except_cluster_specific|

nfs|cifs|ftp|backup|replication|storage_schedules|storage_fs_alert|

compress_schedules|defrag_schedules|storage_dedup|smartio }

`config import remote` *URL [config_type]*

{ network|admin|all|report|system|cluster_specific|all_except_cluster_specific|

nfs|cifs|ftp|backup|replication|storage_schedules|storage_fs_alert|

compress_schedules|defrag_schedules|storage_dedup|smartio }

`config delete` *file_name*

`config list`

19.3.2 DESCRIPTION

The system `config` command imports or exports the Veritas Accessconfiguration settings. The default value: `all` will be setted if you leave `config_type` as blank.

`Config Name Config information`

`========== ==========`

`admin:` List of users, passwords. This includes cifs local users/groups also.

`network:` DNS, LDAP, NIS, netgroup, nsswitch settings (does not include IP).

`report:` Report settings.

`system:` NTP settings, timezone and system options like cfsmount_ontimeout,dmpio,dmptune,nfsd,ninodes,tunefstab,vxtune.

`all:` All the configuration information.

`all_except_cluster_specific:` All except the cluster-specific configuration.

`nfs:` NFS settings.

`ftp:` FTP settings.

`backup:` NBU client and NDMP configuration

`cifs:` CIFS settings.

`replication:` Replication settings.

`storage_schedules:` Imports dynamic storage tiering (DST) and automated snapshot schedules.

`cluster_specific:` Public IP addresses, virtual IP addresses, and console IP address. The network connection to the console server may be lost after importing these settings. If this happens, reconnect after importing this configuration.

`storage_fs_alert`: File system alert settings.

`compress_schedules`: Compression schedules.

`defrag_schedules`: Defragmentation schedule.

`storage_dedup`: File system deduplication settings.

`smartio`: SmartIO settings.

19.3.3 OPTIONS

`config export local` *file_name*

Save the configuration in a local file.

`config export remote` *URL*

Save the configuration on a remote machine. Exported File is named export.tar.gz if no file name is specified in the URL. `Note`: When exporting files with ftp URL, the given path is considered relative to the current working directory of the session. For example, to get the file named README from your home directory on your ftp site, use:

```
ftp://user:passwd@my.site.com/README
```

But if you want to get the README file from the root directory of the site, you need to specify the absolute file name:

```
ftp://user:passwd@my.site.com//README
```

(i.e. with an extra slash in front of the file name.)

`config import local` *file_name [config_type]* { network | admin | all | report | system | cluster_specific | all_except_cluster_specific | nfs | cifs | ftp | backup | replication | storage_schedules | storage_fs_alert | compress_schedules | defrag_schedules | storage_dedup | smartio }

Import the specified configuration saved in a local file. `Note` For importing cifs local users/groups you have to import `admin` module also.

Warning: The `config import` command overwrites the existing configuration settings.

`config import remote` *URL [config_type]* { network | admin | all | report | system | cluster_specific | all_except_cluster_specific | nfs | cifs | ftp | backup | replication | storage_schedules | storage_fs_alert | compress_schedules | defrag_schedules | storage_dedup | smartio }

Import the specified configuration from a remote machine. `Note` For importing cifs local users/groups you have to import `admin` module also. `Note`: When exporting files with ftp URL, the given path is considered relative to the current working directory of the session. For example, to get the file named README from your home directory on your ftp site, use:

```
ftp://user:passwd@my.site.com/README
```

But if you want to get the README file from the root directory of the site, you need to specify the absolute file name:

```
ftp://user:passwd@my.site.com//README
```

(i.e. with an extra slash in front of the file name.)

`config delete` *file_name*

Delete the locally-saved configuration file.

`config list`

View the list of locally-saved configuration files.

19.3.4 EXAMPLES

Export the configuration into a file with the name 2014_July_20.

```
System> config export local 2014_July_20
```

Import only the network configuration.

```
System> config import local 2014_July_20 network
Backup of current configuration was saved as 20140715051522
network configuration was imported
Configuration files are replicated to all the nodes
```

Export (or save) the configuration onto a remote machine.

```
System> config export remote
scp://root@10.209.105.138:/root/2014_July_20.tar.gz
or  ftp://root@10.209.105.138://home/user1/2014_July_20.tar.gz
Password: *******
```

Import the report configuration which is on a remote machine.

```
System> config import remote
ftp://user1@server.com//home/user1/2014_July_20.tar.gz report
Password: *******
```

19.3.5 SEE ALSO

upgrade(1)

19.4 disable

19.4.1 SYNOPSIS

`guidisable`

19.4.2 DESCRIPTION

The `guidisable` command disables the GUI.

19.4.3 OPTIONS

`guidisble` Disable the GUI.

19.4.4 EXAMPLES

Disable the GUI.

```
System> guidisable
Force stopping vamgmt service.
vamgmt service is not running.
```

19.4.5 SEE ALSO

guidisable(1)

19.5 enable

19.5.1 SYNOPSIS

`guienable`

19.5.2 DESCRIPTION

The `guienable` command enables or starts the GUI.

19.5.3 OPTIONS

`guienable` Start or enable the GUI.

19.5.4 EXAMPLES

Enable the GUI console.

```
System> guienable
Start vamgmt service vamgmt...
vamgmt service is running.
```

19.5.5 SEE ALSO

guienable(1)

19.6 status

19.6.1 SYNOPSIS

`guistatus`

19.6.2 DESCRIPTION

The `guistatus` command is used to check the status of the GUI.

19.6.3 OPTIONS

`guistatus` Check the status of the GUI console.

19.6.4 EXAMPLES

Check the status of the GUI console.

```
System> guistatus
Checking service vamgmt...
vamgmt service is not running.
```

19.6.5 SEE ALSO

guistatus(1)

19.7 kms

19.7.1 SYNOPSIS

`kms config list`

`kms config status`

`kms config server` *server_ip server_port*

`kms config delete`

`kms certificate generate`

`kms certificate import_server_cert`

`kms certificate import_client_key`

`kms certificate import_client_cert`

19.7.2 DESCRIPTION

The system `kms` commands are used to register or unregister the cluster with the KMS server.

19.7.3 OPTIONS

`kms config list` Display the registered KMS server details.

`kms config status` List the status of SSL certificate files that are configured on the cluster.

`kms config server` *server_ip server_port* Register the KMS server with cluster.

`kms config delete` Delete the KMS server details.

`kms certificate generate` Generate SSL self-signed keypair on cluster. It is advised that cluster nodes' time settings are synchronized with NTP server before generating self-signed certificates.

`kms certificate import_server_cert` Import the KMS servers public key.

`kms certificate import_client_key` Import the SSL key file of the cluster that is used in CSR.

`kms certificate import_client_cert` Import the CA signed public key of the cluster.

19.7.4 EXAMPLES

Display registered KMS server details.

```
System> kms config list
Configured KMS server host: 10.182.65.65
Configured KMS server port: 5696
```

Register KMS server 10.182.65.65 on cluster. Note: We need to import and generate SSL keypair with KMS server certificate file import before runinng "kms config server" command. KMS servers with OASIS KMIP 1.1 specification are supported.

```
System> kms config server 10.182.103.113 5696
100% [#] KMS server register with cluster nodes.
ACCESS kms SUCCESS V-288-0 KMS server registered successfully.
```

Display the configuration status of KMS server communication related SSL certificates.

```
System> kms config status
ACCESS kms SUCCESS V-288-0 KMS server certificate file imported.
```

To unregister KMS server on cluster.

```
System> kms config delete
ACCESS kms SUCCESS V-288-0 Successfully unregistered KMS server.
```

To generate a self-signed SSL keypair for the cluster.

```
System> kms certificate generate
ACCESS kms SUCCESS V-288-0 Self-signed certificates successfully generated.
```

To import the KMS server's public key on the cluster. **Note:** We need to generate/import the SSL keypair for the cluster before this operation.

```
System> kms certificate import_server_cert
================================================================
       Please enter KMS Server certificate details
================================================================

Enter KMS Server public key certificate identified by
-----BEGIN CERTIFICATE-----
... (KMS Server certificate in base64 encoding) ...
-----END CERTIFICATE-----


-----BEGIN CERTIFICATE-----
MIICyDCCAbCgAwIBAgIHAdBRB9i6bzANBgkqhkiG9w0BAQsFADATMREwDwYDVQQD
EwhJQk0gU0tMTTAeFw0xNzAzMDExMjM1NDJaFw00MTEwMjExMjM1NDJaMBMxETAP
BgNVBAMTCElCTSBTS0xNMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA
uu+FjPQEePp92wugJhV4I2V/ADyfgtxD/kcaCWf3pCfKZlLp0Py9F6TSnDIrvZHl
xI6aL9UUieR9f4xjjoNt2yEJ/nWxu/9eu3yqGDkf7GyEoo8aWDki+ewGSCmaI1xC
8lE0+gNbpaDPl7Xbk9WqMELHo2+6+wVmie9Kx6VuQsYvF1ZIeOmdw8Zjcxc8uxWj
b8PjrqHXrVb9DEAP/kAe/KdkwRMDSEhzBUz/Of6ZVMCivrBI8ItqLS3wzWwAGf+K
L7+D8iVmrnPmt9QAcAl2EWhThNWAc/IR+88l+bdTfsH6L5d7fYmisLv2pDg0VWkN
8fvM9JeWDxmUSaO8SkzHHQIDAQABoyEwHzAdBgNVHQ4EFgQUnvpjLMIMq8lzG6rN
n4KVZhmWW6YwDQYJKoZIhvcNAQELBQADggEBAJPQ4xoOgWDadQjRbXBzl0bxg8fB
aH3ka5OBSQJOHo4eVg3LFlQ3Fd8D6l9We7M03KxURKp7S1u2HyUjJCPZbfS2X6IR
qn8ynY2GjAuFqp0sAIFAiE7LSN4IUkqSiGCESiQQNWpTF9yTJo8KaWuUBqw1onFl
YAu/C0csx/nzYdbSMY4qsrL+CHmM5ALHxh9PTQfndvtngge2DEXZr4nwxsS8athE
S4tZ9rJ8KqVuzrpwgsMVn5x913tU2+UmFJacTm/BvKAIWC8LHQihCOdDUDRcSf48
b2SzdUF3SCMKHMETKcFEB7s5mm3Uo9NP+NLRM6RNFwwg1nyEGsFVyQ/ZwXk=
-----END CERTIFICATE-----

ACCESS kms SUCCESS V-288-0 Successfully imported KMS server certificate.
```

To import the cluster's SSL key file used during CSR creation.

```
System> kms certificate import_client_key
================================================================
       Please enter client keyfile details
```

```
================================================================

Enter client's private key identified by
-----BEGIN RSA PRIVATE KEY-----
... (Client key in base64 encoding) ...
-----END RSA PRIVATE KEY-----


-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAncilNq/YBjwEervKIMFwjwExdePxlJjb5Ym0MVG9W+TmGwvW
4e47WCXCubZp93jfVQ4HfjR+mzI4ARcBrFbh9hivuwrZj8NF2IzUUxHPeuGlQMmZ
H8bIzA5mkZQL+4WAUKblZnDZrQCayBYHYJZk4Xo3GG0qZeU53gIni1QaNToDsYQB
hKlftjb+A/KB5GsfZJJoRdcSnd5az16+sxa//r1bKL7J76NIYJRR6x2UXHrXTvG9
Pq+ciCV8mfTjW4T+oBylCb8o3bUxGdFlBtEFwhiDZwtO/gCkL2qiCbQxcXGdWjTc
0+BeUwN4wJwnXbcJ4qwdAKBZANQYQ2MWCPId1QIDAQABAoIBADKLTfGAzxDgNQck
FRH9JZOJ7FA5nu/dL8QOIr0R1DkV0GilmEZzzGrmXcSkHLn77H8QqXHSWyRYJ/yh
ZBXTFzuAmPqv9RXKlhtSscmSkQZqNkfUzMqTIUvM6SqFZHlEQOCVNjNkvzZeVTb3
Z5GXW+WnAvM+NTDDB5CI3CYT4PlfhXy+Glij6dlp0rhyJsun0qpjKoFqQzLpGhXm
IgAnfBZ8nDQbxODM9N9rb30ry+j/f3SlNEX6hhNkbD1lbT/uQmz2RSceFzkLned0
JG7NMPZQQqhxu2l0GPAUoLqopjp4JL85Xk/a4rXBRAL4tN1SRX9B803TkXXuaWU7
qktUelkCgYEAy6jtEqWjYhCxpSOVsl1LNaPkY4B1LBGktBUkhyCcuvFZ5AohCuNi
/lQNkk18xgdnbDoycB7v0s9CG5S4Po1ydOg9cfXDmr8kf98Bip9HjorPVmE6RZDT
la8TkOcm5GtJIgXsnNaLUuxKws/i4SZ1N4vxhhQcgU1OF1odHT9kit8CgYEAxlV3
HjL2pRPt0snRRbCNXLb5tTXmH9hWE1CNSJdoxkc0KzD0zJOCMHxdmFHEG0Q7JANN
RssKStt21TTrWdgmjlyH1zEcbQkX9eSf2+3rQ53+K9Gx3u2iFum5wUXyR8qZKdtd
kha61TQXrhvhr+NsBvMDbe42Jr+lU2t97GkW4csCgYBKrkdA7zxaWYTK6aGbJlrN
8KtvUUumPVIINziN1IekhZxQ4uX6+Cb03B2d6Iw4kb57EVwtgSqwQxAazQ5C4dt2
wzLE9zojJLRdr4kBLCcxPfWJ111m5RSxbuHd4OWCHVVUBJgfawzr+3nutgycQ134
YHjQkc9iGtbn+UNz1isoQQKBgCMjYy5DgzUt+bWdsejx5ammdUkomnPL5TwYyl8x
DAwM824vhz9zZt5mOkLRQbg+92di+l7vfAxVI5GUhZnuVK5mMF7swYeD0+3pqnTR
ixv4st5akXPYkB4VHKbkWI9lkos65TILuuApEYT0RaDe3vXAOjxQtuFYlNbZtX6c
A8TPAoGAZ7YvcRluDT7mRNWa+EOTslJ1/POvtvnH5Yw/+n+C0vpBiyy1Zxaa0zyq
7R/aD9ejEZEIo7n/tWPCgvI6VvBz0+Jz6UIm2p1TtPuJcMjxP4Vnnbp7qfUmhC7M
WXVRlyTgSxqYhexAkG4SaQ0nPTnrnfxL1f6kMoHob+qeL705NDA=
-----END RSA PRIVATE KEY-----

ACCESS kms SUCCESS V-288-0 Client key file successfully imported.
```

To import the cluster's SSL certificate file signed by CA.

```
System> kms certificate import_client_key
================================================================
       Please enter KMS Server certificate details
================================================================

Enter KMS Server public key certificate identified by
-----BEGIN CERTIFICATE-----
... (KMS Server certificate in base64 encoding) ...
-----END CERTIFICATE-----


-----BEGIN CERTIFICATE-----
MIICyDCCAbCgAwIBAgIHAdBRB9i6bzANBgkqhkiG9w0BAQsFADATMREwDwYDVQQD
EwhJQk0gU0tMTTAeFw0xNzAzMDExMjM1NDJaFw00MTEwMjExMjM1NDJaMBMxETAP
BgNVBAMTCElCTSBTS0xNMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA
uu+FjPQEePp92wugJhV4I2V/ADyfgtxD/kcaCWf3pCfKZlLp0Py9F6TSnDIrvZHl
xI6aL9UUieR9f4xjjoNt2yEJ/nWxu/9eu3yqGDkf7GyEoo8aWDki+ewGSCmaI1xC
8lE0+gNbpaDPl7Xbk9WqMELHo2+6+wVmie9Kx6VuQsYvF1ZIeOmdw8Zjcxc8uxWj
```

```
b8PjrqHXrVb9DEAP/kAe/KdkwRMDSEhzBUz/Of6ZVMCivrBI8ItqLS3wzWwAGf+K
L7+D8iVmrnPmt9QAcAl2EWhThNWAc/IR+88l+bdTfsH6L5d7fYmisLv2pDg0VWkN
8fvM9JeWDxmUSaO8SkzHHQIDAQABoyEwHzAdBgNVHQ4EFgQUnvpjLMIMq8lzG6rN
n4KVZhmWW6YwDQYJKoZIhvcNAQELBQADggEBAJPQ4xoOgWDadQjRbXBzl0bxg8fB
aH3ka5OBSQJOHo4eVg3LFlQ3Fd8D6l9We7M03KxURKp7S1u2HyUjJCPZbfS2X6IR
qn8ynY2GjAuFqp0sAIFAiE7LSN4IUkqSiGCESiQQNWpTF9yTJo8KaWuUBqw1onFl
YAu/C0csx/nzYdbSMY4qsrL+CHmM5ALHxh9PTQfndvtngge2DEXZr4nwxsS8athE
S4tZ9rJ8KqVuzrpwgsMVn5x913tU2+UmFJacTm/BvKAIWC8LHQihCOdDUDRcSf48
b2SzdUF3SCMKHMETKcFEB7s5mm3Uo9NP+NLRM6RNFwwg1nyEGsFVyQ/ZwXk=
-----END CERTIFICATE-----

ACCESS kms SUCCESS V-288-0 Successfully imported KMS server certificate.
```

19.8 license

19.8.1 SYNOPSIS

`license add` *<path_of_key_file>*

`license list`

`license list details`

19.8.2 DESCRIPTION

The `license` commands provide options to perform necessary operations related to licensing. It adds or installs a license. It lists the type of license installed on the cluster. It also gives details of the installed license.

19.8.3 OPTIONS

`license add` ***<path_of_key_file>*** Adds or installs the license using the .slf key file which is provided. The scp protocol is also supported for adding license. The scp path should be specified: scp://user@ip:<path_of_key_file>

`license list` Lists the type of license installed on the cluster. The license type can be trialware (trial license for 60 days), subscription (license with validity of 1 year) or perpetual (permanent license).

`license list details` Lists the details of the license installed on the cluster such as the product name, version, start date, end date, license type, current state etc.

19.8.4 EXAMPLES

Adding license using local **.slf** file.

```
System>license add /home/user1/subscription.slf
ACCESS license SUCCESS V-288-0 License installed successfully on all nodes of the␣
↪cluster.
```

Using scp for installing license.

```
System>license add scp://user1@10.50.148.208:/home/user1/subscription.slf
user1@10.50.148.208's password:
ACCESS license SUCCESS V-288-0 License installed successfully on all nodes of the␣
↪cluster.
```

When license installation fails.

```
System>add /home/user1/key.slf
ACCESS license ERROR V-288-0 Input License file not properly installed on all the␣
↪nodes; Successfully installed on 0 out of 2 nodes; Installation failure on: ['isaA_
↪01', 'isaA_02']
```

When an error occurs while fetching the license information.

```
System>license list
ACCESS license ERROR V-288-0 Could not retrieve license information due to internal␣
↪error
```

When a license is installed and the `license list details` command is used.

```
System>license list details
Product Name    Version   Product Edition   License Meter   License Type   Start Date␣
↪  End Date   State
==============  ========  ================  ==============  ============= ␣
↪===========  =========  ======
Veritas Access  7.2.1     ENTERPRISE        PER-CORE        PERPETUAL      NA        ␣
↪  NA         Active
```

When subscription license is installed on the entire cluster

```
System>license list
Installed License Type
=======================
SUBSCRIPTION

System>license list details
Product Name    Version   Product Edition   License Meter   License Type   Start Date␣
↪  End Date    State
==============  ========  ================  ==============  ============= ␣
↪===========  ==========  ======
Veritas Access  7.3       ENTERPRISE        PER-CORE        SUBSCRIPTION   2017-05-25␣
↪  2018-07-24  Active
```

When the license is installed only on some nodes of the cluster or licensing information from some nodes cannot be fetched as the nodes are down or not reachable.

```
System>license list
Installed License Type
=======================
SUBSCRIPTION

ACCESS license WARNING V-288-0 License is partially installed on this cluster. For␣
↪more details refer to the Veritas Access GUI.

System>license list details
Product Name    Version   Product Edition   License Meter   License Type   Start Date␣
↪  End Date    State
==============  ========  ================  ==============  ============= ␣
↪===========  ==========  ======
Veritas Access  7.3       ENTERPRISE        PER-CORE        SUBSCRIPTION   2017-05-25␣
↪  2018-07-24  Active

ACCESS license WARNING V-288-0 License is partially installed on this cluster. For␣
↪more details refer to the Veritas Access GUI.
```

When a license is not installed on the cluster

```
System>license list
ACCESS license WARNING V-288-0 No license is installed on this cluster.


System>license list details
ACCESS license WARNING V-288-0 No license is installed on this cluster.
```

19.9 more

19.9.1 SYNOPSIS

`more status`

`more enable`

`more disable`

19.9.2 DESCRIPTION

The system `more` command enables, disables, or checks the status of the more filter. If this command is enabled, then it activates the more filter for paging through text one screen at a time. By default, the more filter is enabled on all of the nodes in the cluster. The more disable command deactivates the more filter on all of the nodes in the cluster.

19.9.3 OPTIONS

`more status` Display the status of the more filter.

`more enable` Enable the more filter on all of the nodes in the cluster.

`more disable` Disable the more filter on all of the nodes in the cluster.

19.9.4 EXAMPLES

Show the status of the more filter.

```
System> more status
Status  : Enabled
```

Enable the more filter on all of the nodes.

```
System> more enable
ACCESS more Success V-288-0 more activated on console.
```

Disable the more filter on all of the nodes.

```
System> more disable
ACCESS more Success V-288-0 more deactivated on console.
```

19.10 ntp

19.10.1 SYNOPSIS

`ntp show`

`ntp enable`

`ntp disable`

`ntp servername` *server-name*

`ntp sync` [*server-name*]

19.10.2 DESCRIPTION

The system `ntp` commands start, stop, synchronize, or check the status of the NTP server.

19.10.3 OPTIONS

`ntp show` Display the NTP status and server names.

`ntp enable` Enable NTP on all of the nodes in the cluster.

`ntp disable` Disable NTP on all of the nodes in the cluster.

`ntp servername` ***server-name*** Set the comma-separated list of NTP servers on all of the nodes in the cluster. Use *127.127.1.0* as the server name to set the local clock.

`ntp sync` [***server-name***] Synchronize date with the NTP server on all of the nodes in the cluster. `Note:` NTP should be disabled before running the "ntp sync [server name]" command.

19.10.4 EXAMPLES

Display the NTP status and server name.

```
System> ntp show
Status      :    Disabled
Server Name:    127.127.1.0
```

Set `10.10.10.10` and `20.20.20.20` as the NTP server on all of the nodes.

```
System> ntp servername 10.10.10.10,20.20.20.20
ACCESS System INFO V-288-0 ['10.10.10.10', '20.20.20.20'] has been added into NTP␣
↪server.
```

Enable NTP on all of the nodes.

```
System> ntp enable
Enabling the NTP server.  Done
```

Disable NTP on all of the nodes.

```
System> ntp disable
Disabling the NTP server.  Done
```

Sync date with NTP server on all of the nodes.

```
System> ntp sync
Date is synchronized on all nodes.
```

Sync date with specific NTP server on all of the nodes.

```
System> ntp sync 30.30.30.30
Sync from ['30.30.30.30']...
Date is synchronized on all node.
```

19.10.5 SEE ALSO

clock(1)

19.11 option

19.11.1 SYNOPSIS

`option show nfsd`

`option show cfsmount_ontimeout`

`option show dmpio`

`option show ninodes`

`option show tunefstab`

`option show dmptune`

`option show vxtune`

`option show vxfs`

`option modify nfsd` *number* [*nodename*]

`option modify cfsmount_ontimeout` *value*

`option modify dmpio` {enclosure *enclr_name* | arrayname

array_name | arraytype *{A/A|A/P|...}* }

iopolicy{adaptive | adaptiveminq | balanced

minimumq | priority | round-robin | singleactive}

`option modify ninodes` *{ number | Auto }*

`option modify tunefstab` *write_throttle value*

`option modify dmptune` {dmp_path_age *value* | dmp_health_time *value*}

`option modify vxtune` *volpagemod_max_memsz value*

`option modify vxfs` *vx_timelag value*

19.11.2 DESCRIPTION

The system `option` command is used to display and configure the tunable parameters. The description of the DMP I/O policy is as follows:

adaptive: In SAN environments, determines the paths that have the least delay, and schedules I/O on paths that are expected to carry a higher load. Priorities are assigned to the paths in proportion to the delay.

adaptiveminq: Similar to the adaptive policy, except that I/O is scheduled according to the length of the I/O queue on each path. The path with the shortest queue is assigned the highest priority.

balanced: Takes the track cache into consideration when balancing I/O across paths.

minimumq: Uses a minimum I/O queue policy. I/O is sent on paths that have the minimum number of I/O requests in the queue. This policy is suitable for low-end disks or JBODs where a significant track cache does not exist. This is the default policy for Active/Active (A/A-A) arrays.

priority: Assigns the path with the highest load-carrying capacity as the priority path. This policy is useful when the paths in a SAN have unequal performance, and you want to enforce load-balancing manually.

round-robin: Sets a simple round-robin policy for I/O. This is the default policy for Active/Passive (A/P) and Asynchronous Active/Active (A/A-A) arrays.

singleactive: I/O is channeled through the single active path. The optional attribute use_all_paths controls whether the secondary paths in an Asymmetric Active/Active (A/A-A) array are used for scheduling I/O requests in addition to the primary paths. The default setting is no, which disallows the use of the secondary paths.

19.11.3 OPTIONS

`option show nfsd` Display the number of NFS daemons for each node in a cluster.

`option show cfsmount_ontimeout` Display cfsmount online timeout.

`option show dmpio` Display the type of iopolicy corresponding to enclosure, arrayname, arraytype for each node.

`option show ninodes` Display the global inode cache size.

`option show tunefstab` Display the global value of the write_throttle parameter.

`option show dmptune` Display the value of the dmptune attribute.

`option show vxtune` Display the tunable values of the volume.

`option show vxfs` Display the tunable parameters for VxFS.

`option modify nfsd` ***number*** **[*nodename*]** Modify the number of NFS daemons. The range for the number of daemons is from 1 to 512.

Warning: The `modify nfsd` command overwrites the existing configuration settings.

`option modify cfsmount_ontimeout` ***Value*** Modify cfsmount online timout. The range from 300 to 9000.

`option modify dmpio` **{enclosure** ***enclr_name*** **| arrayname**

array_name **| arraytype** ***{A/A|A/P|...}*** **}** iopolicy{adaptive | adaptiveminq | balanced minimumq | priority | round-robin | singleactive}

Modify the dmpio policy corresponding to enclosure, arrayname, arraytype. Warning: Check the sequence before modifying the I/O policy. The policies need to be applied in the following sequence: arraytype, arrayname, and enclosure. The enclosure-based modification of the I/O policy overwrites the I/O policy set using the arrayname and the arraytype for that particular enclosure. In turn, the arrayname-based modification of the I/O policy overwrites the I/O policy set using the arraytype for that particular arrayname.

`option modify tunefstab` ***write_throttle value*** Modify the value of the write_throttle parameter.

`option modify dmptune` ***dmp_health_time value*** Modify the value of dmp_health_time. This attribute sets the time in seconds for which a path must stay healthy. If a path's state changes back from enabled to disabled within this time period, DMP marks the path as intermittently failing, and does not re-enable the path for I/O until dmp_path_age seconds elapse. The default value of dmp_health_time is 60 seconds. A value of 0 prevents DMP from detecting intermittently failing paths.

`option modify dmptune` ***dmp_path_age value*** Modify the value of dmp_path_age. The time for which an intermittently failing path needs to be monitored before DMP marks the path as healthy and once again attempts to schedule I/O requests on it. The default value is 300 seconds. A value of 0 prevents DMP from detecting intermittently failing paths.

`option modify ninodes` ***{ number | Auto }*** Modify the global inode cache size. The range for inode cache size is from 10000 to 8000000. Set to Auto to enable autoreset by VXFS.

Warning: The `modify ninodes` command requires a cluster-wide reboot.

option modify vxtune *volpagemod_max_memsz value* Modify the value of volpagemod_max_memsz. This is the maximum memory measured in kilobytes that is allocated for cache object metadata. The default value for volpagemod_max_memsz is set to 131072KB. The value that should be used is determined by the total size of volumes for which instant rollbacks are to be taken. The following formula can be used to calculate the required value of volpagemod_max_memsz:

size_in_KB=6 * (total_filesystem_size_in_GB) * (64/region_size_in_KB)

region_size can be set to 256KB by default for large filesystems.

option modify vxfs *vx_timelag value* Modify the values of vx_ifree_timelag and vx_iclean_timelag .

vx_ifree_timelag

VxFS maintains an inode free list. If you configure the vx_ifree_timelag value as 30 seconds, the freelist is scanned every 30 seconds.

vx_iclean_timelag

This is the minimum time that an inode must be in the inode free list before the system reclaims it.

VALUE Specify an integer value (Unit: Seconds)

19.11.4 EXAMPLES

Show the value of the number of NFS daemons.

```
System> option show nfsd
NODENAME             NUMBER_DAEMONS
--------             --------------
sfs_01               66
```

Show cfsmount online timeout.

```
System> option show cfsmount_ontimeout
Resource            OnlineTimeout
--------          --------------
cfsmount               300
```

Show the value of dmpio policy corresponding to enclosure, arrayname, arraytype.

```
System> option show dmpio
NODENAME             TYPE                 ENCLR/ARRAY          IOPOLICY
--------             ---------            --------             --------
sfs_01               enclosure            disk                 Balanced
sfs_01               enclosure            aluadisk0            Priority
sfs_01               arraytype            A/A                  balanced
sfs_02               enclosure            disk                 Balanced
sfs_02               enclosure            aluadisk0            Priority
sfs_02               arraytype            A/A                  balanced
```

Show the value of the global inode cache size.

```
System> option show ninodes
INODE_CACHE_SIZE
----------------
565580
```

Show the value of the tunefstab parameters.

```
System> option show tunefstab
NODENAME             ATTRIBUTE            VALUE
--------             ---------            -----
sfs_01               write_throttle       0
```

Show the value of the dmptune parameters.

```
System> option show dmptune
NODENAME             ATTRIBUTE            VALUE
--------             ---------            -----
sfs_01               dmp_path_age         57
sfs_01               dmp_health_time      44
```

Show the value of the vxtune parameters.

```
System> option show vxtune
NODENAME          TUNABLE                              VALUE(KB)
--------          -------                              ---------
sfs_01            volpagemod_max_memsz                     12288
sfs_02            volpagemod_max_memsz                     12288
```

Modify the number of NFS daemons on all the nodes in a cluster.

```
System> option modify nfsd 97
```

Modify cfsmount online timeout.

```
System> option modify cfsmount_ontimeout 400
```

Modify the dmpio policy, enclosure, and diskname.

```
System> option modify dmpio enclosure Disk Balanced
```

Modify the global inode cache size.

```
System> option modify ninodes 2000343
```

Enable inode cache size autoreset.

```
System> option modify ninodes Auto
```

Modify the write_throttle parameter of tunefstab.

```
System> option modify tunefstab write_throttle 20003
```

Modify the dmp_path_age parameter of dmptune.

```
System> option modify dmptune dmp_path_age 40
```

Modify the dmp_health_time parameter of dmptune.

```
System> option modify dmptune dmp_health_time 50
```

Modify the volpagemod_max_memsz parameter of vxtune

```
System> option modify vxtune volpagemod_max_memsz 12288
VXTUNE    - set volpagemod_max_memsz as 12288(KB)

System> option modify vxtune volpagemod_max_memsz 54533443234
VXTUNE    - VxVM vxtune ERROR V-5-1-18394 Tunable value 54533443234 for tunable␣
↪volpagemod_max_memsz is out of range [0 - 4294967295]
```

19.11.5 SEE ALSO

clock(1), ntp(1), stat(1), swap(1)

19.12 stat

19.12.1 SYNOPSIS

`stat sys` [*node*]

`stat dmp` [*node*]

`stat fsio` [*fsname*]

`stat cluster`

`stat all` [*node*]

`stat rdma` [*node*]

19.12.2 DESCRIPTION

System `stat` command displays the system, DMP, and process-related node-wide statistics. The load in the displayed output is the load from the last 1, 5, and 15 minutes. Intr is the total number of interrupt counts, and ctxt is the total number of context switches that occurred after a reboot.

The stat command cluster option displays I/O and network throughput at the entire cluster level.

19.12.3 OPTIONS

`stat sys` [*node*]

Display system-related statistics. *node* : name of a node in the cluster.

`stat dmp` [*node*]

Display DMP-related statistics. *node* : name of a node in the cluster.

`stat fsio` [*fsname*]

Display filesystem IO statistics. *fsname* : name of the filesystem.

`stat cluster` Display cluster IO and network throughput.

`stat all` [*node*]

Display system and DMP-related statistics of all the nodes or a node in the cluster at a time. *node* : name of a node in the cluster.

`stat rdma` [*node*]

Display RDMA statistics of all the nodes or specific node(s) in the cluster at a time. *node* : name of node(s) in the cluster.

19.12.4 EXAMPLES

Display system-related statistics of node1.

```
System> stat sys node1
Gathering statistics...
GEN STAT :::
============
CPU Idle   : 99.450     Load 1  : 1.020
```

```
CPU System : 0.050      Load 5  : 1.020
CPU User   : 0.050      Load 15 : 1.010
ctxt       : 184128974    intr    : 392690761  processes(R) : 1009379

MEM STAT :::
============
Mem Total : 4040096    Swap total : 1052248     %Mem used  : 49.610%
Mem Used  : 2004100    Swap Used  : 0           %Swap Used : 0.000%
Mem Free  : 2035996    Swap Free  : 1052248     %Total use : 49.610%
NET STAT :::
============
IFACE      rxmb/s    txmb/sec  rxerr/s   txerr/s   coll/s    rxdrop/s  txdrop/s  dev_
↪speed Mode
------     -----     --------  -------   -------   ------    --------- -------   -----
↪---- -----
pubeth0    0.000     0.000     0.000     0.000     0.000     0.000     0.000     1000␣
↪     Full
priveth0   0.000     0.000     0.000     0.000     0.000     0.000     0.000     1000␣
↪     Full

DISK STAT :::
=============
                      OPERATIONS          BLOCKS           AVG TIME(ms)
TYP NAME              READ     WRITE      READ     WRITE   READ  WRITE
dm  Disk_0               0         0         0         0    0.0    0.0
dm  Disk_1               0         0         0         0    0.0    0.0
dm  Disk_2               0         0         0         0    0.0    0.0
dm  Disk_3               0         0         0         0    0.0    0.0
```

Display consolidated system-related statistics of all running nodes in the cluster.

```
System> stat sys
Gathering statistics...
Node       %cpu_util %mem_util av_read_time(ms) av_write_time(ms) rx_av_pubeth(mb/s)␣
↪tx_av_pubeth(mb/s)
----       --------- --------- ---------------- ----------------- ----------------- --
↪----------------
node1     0.800     49.680     0.000            0.000             0.000              0.
↪000
node2     0.150     50.110     0.000            0.000             0.000              0.
↪000
```

Display DMP-related statistics of node1.

```
System> stat dmp node1
DMP STAT :::
============
                         OPERATIONS             KBYTES            AVG TIME(ms)
PATHNAME              READS     WRITES      READS     WRITES     READS   WRITES
sda                     978          0         23          0 65.695652 0.000000
sdf                     477        155      98631       1028  0.285174 0.715953
sdd                     457        186      92455       1264  0.266043 0.471519
sdg                    3043         63      67010        502  2.804343 0.364542
sde                    6096        147      95382       1003  4.512057 0.664008
sdb                     159         49      29957        440  0.449411 0.554545
sdc                    1108         80      36071        326  0.208395 0.478528
```

Display consolidated DMP-related statistics of all the running nodes in the cluster.

```
System> stat dmp
Node        Path        Av_read_time(ms) Av_write_time(ms)
----        ----        ---------------- -----------------
sfs1_1      sda           65.696            0.000
sfs1_1      sdf           0.285             0.716
sfs1_1      sdd           0.266             0.472
sfs1_1      sdg           2.804             0.365
sfs1_1      sde           4.512             0.664
sfs1_1      sdb           0.449             0.555
sfs1_1      sdc           0.208             0.479
sfs1_2      sda           28.857            0.000
```

Display filesystem IO statistics of all running nodes in the cluster.

```
System> stat fsio testfs1
                              OPERATIONS              BLOCKS          AVG TIME(ms)
NAME                         READ      WRITE       READ      WRITE    READ   WRITE
----                         ---------------       ---------------    ------------
Node: sfs1_0
==============
testfs1_tier1                 532        206       2336       7486    1.17    9.98
testfs1_tier2                   1          1          2         16    4.00    8.00
Node: sfs1_1
==============
testfs1_tier1                 369        168       1970       3218    2.12    7.67
testfs1_tier2                   1          0          2          0    0.00    0.00
```

Display system and DMP-related statistics of node1.

```
System> stat all node1
```

Display system and DMP-related statistics of all nodes in the cluster.

```
System> stat all
```

Display cluster wide Network and IO throughput.

```
System> stat cluster
Gathering statistics...
Cluster wide statistics::::
============================
IO throughput :: 0
Network throughput :: 1.205
```

Display RDMA network statistics.

```
System> stat rdma
Gathering statistics...
TAG-NAME   MODE   PRIORITY   MTU   BROADCAST     TxPKTS(M)   TxBYTES(MB)   RxPKT
=========  =====  =========  ====  ============  ==========  ============  =====
priveth0   rdma   hipri      8192  172.16.0.255  1.45        106.35        1.34
priveth1   rdma   hipri      8192  172.16.1.255  1.45        106.35        1.34
```

Display RDMA network statistics for specific node(s).

```
System> stat rdma rdma_01,rdma_02
Gathering statistics...
NODE     TAG-NAME   PROTOCOL   NODE-RANGE   LINK-TYPE   UDP-PORT   MTU   IP ADDRESS
↪BCAST-ADDRESS
```

```
=======  =========  =========  ===========  ==========  =========  ====  ===========
↪===============
rdma_02  priveth0   udp        -            rdma        51001      -     172.16.0.4
↪172.16.0.255
rdma_02  priveth1   udp        -            rdma        51002      -     172.16.1.4
↪172.16.1.255
rdma_01  priveth0   udp        -            rdma        51001      -     172.16.0.3
↪172.16.0.255
rdma_01  priveth1   udp        -            rdma        51002      -     172.16.1.3
↪172.16.1.255
```

NOTE: MTU: '-' is the default, which has a value of 8192. NODE-RANGE: '-' indicates all cluster are to be configured for this link.

19.13 webservices

19.13.1 SYNOPSIS

`webservices server` [start|stop|status]

`webservices SSL certificate` *certificate_path*

`webservices SSL disable`

`webservices SSL enable`

`webservices user password`

19.13.2 DESCRIPTION

The system `webservices` commands allow you to maintain the REST server. You can start, stop, or display the status of the REST server. You can also enable, disable, or set the SSL certificate for the REST server. You can also change the password of the REST server user.

19.13.3 OPTIONS

certificate Certificate file path and name. Make sure to set the correct read permissions for the certificate file.

`server` [start|stop|status] Start, stop, or display the status of the REST server.

`SSL certificate` *certificate_path* Set the certificate file of SSL for the REST server.

`SSL disable` Disable SSL for the REST server.

`SSL enable` Enable SSL for the REST server.

`user password` Set the user password. Currently, Veritas Access supports only one user, the manila user.

19.13.4 EXAMPLES

Start the REST server.

```
System> webservices server start
ACCESS REST SUCCESS V-288-1397 REST server is started.
```

Display the status of the REST server.

```
System> webservices server status
REST Server : http://10.200.114.252:8088
Certificate : /opt/VRTSnas/conf/ssl.pem
User        : manila
SSL         : disable
Status      : ONLINE
```

Stop the REST server.

```
System> webservices server stop
ACCESS REST SUCCESS V-288-1397 REST server is stopped.
```

Set the certificate file for the REST server.

```
System> webservices SSL certificate /opt/VRTSnas/conf/ssl.pem
ACCESS REST SUCCESS V-288-1397 Set REST server certificate successfully.
```

Disable SSL for the REST server.

```
System> webservices SSL disable
ACCESS REST SUCCESS V-288-1397 REST server SSL disabled.
```

Enable SSL for the REST server.

```
System> webservices SSL enable
ACCESS REST SUCCESS V-288-1397 REST server SSL enabled.
```

Set user "manila" password.

```
System> webservices user password
Changing password for manila
Old password:
New password:
Re-enter new password:
ACCESS REST SUCCESS V-288-1397 REST server user password changed.
```

19.13.5 SEE ALSO

20

Target Commands

20.1 target

20.1.1 SYNOPSIS

`iscsi service` start|stop|status

`iscsi target portal add` *portal-ip*

`iscsi target portal del`

`iscsi target create` *targetname*

`iscsi target destroy` *targetname*

`iscsi target list` [*targetname*]

`iscsi target status` *target-name*

`iscsi lun create` *lun-name lunid target-name size*

`iscsi lun destroy` *lunid target-name*

`iscsi lun list` [*targetname*]

`iscsi target store add` *fs-name target-name*

`iscsi target store delete` *fs-name target-name*

`iscsi target map add` *initiator-name target-name*

`iscsi target map delete` *initiator-name target-name*

`iscsi target auth incominguser add` *username target-name*

`iscsi target auth incominguser delete` *username target-name*

20.1.2 DESCRIPTION

Veritas Access can be configured as an iSCSI target to serve block storage. The iSCSI target service is hosted in active-passive mode in the Veritas Access cluster. Once configured, the cluster is available to any standard iSCSI initiator over a portal IP. You can perform the following functions on an iSCSI target:

1. Starting and stopping of the iSCSI target service

2. Addition and deletion of targets using the `iscsi target` commands
3. Addition and deletion of LUNs using the `iscsi lun` command
4. Map and un-map initiators using the `iscsi target map` commands
5. Addition and deletion of users to set up CHAP authentication

20.1.3 OPTIONS

`iscsi service` **start|stop|status** Start, stop or display the status of the iSCSI target service.

`iscsi target portal add` ***portal-ip*** Add the portal address for an iSCSI target service.

`iscsi target portal del` Delete the portal address for an iSCSI target service.

`iscsi target create` ***targetname*** Create an iSCSI target.

`iscsi target destroy` ***targetname*** Destroy a specific iSCSI target.

`iscsi target list` [***targetname***] List all or specific iSCSI targets.

`iscsi target status` ***target-name*** Check the status of a specific iSCSI target.

`iscsi lun create` ***lun-name lunid target-name size*** Create a LUN with the specified name, LUN ID and size on the specified file system.

`iscsi lun destroy` ***lunid target-name*** Destroy a specific LUN.

`iscsi lun list` [***targetname***] List details of all the LUNs present in all or a specific target.

`iscsi target store add` ***fs-name target-name*** Map file system with specified iSCSI target.

`iscsi target store delete` ***fs-name target-name*** Remove the file system from a specified iSCSI target.

`iscsi target map add` ***initiator-name target-name*** Map the iSCSI initiator with specified iSCSI target.

`iscsi target map delete` ***initiator-name target-name*** Remove the iSCSI initiator mapped to a specified iSCSI target.

`iscsi target auth incominguser add` ***username target-name*** Create an incoming user and bind the account to a specified, existing iSCSI target.

`iscsi target auth incominguser delete` ***username target-name*** Remove an incoming user and unbind the account from its corresponding iSCSI target.

20.1.4 EXAMPLES

Start the iSCSI target service. If the iSCSI target service is already started, the start command clears the faults, if any, and then tries to start the iSCSI target service:

```
Target> iscsi service start
ACCESS Target SUCCESS V-288-0 iSCSI Target service started
```

Stop the iSCSI target service:

```
Target> iscsi service stop
ACCESS Target SUCCESS V-288-0 iSCSI Target service stopped
```

Check the status of iSCSI target service:

```
Target> iscsi service status
Portal IP :- 192.168.10.33
Node    Status
=====   ========
VA_01   OFFLINE
VA_02   OFFLINE
```

Add the portal address for an iSCSI target service:

```
Target> iscsi target portal add 192.168.10.32
ACCESS Target SUCCESS V-288-0 Portal add successful. Start the iSCSI service for the␣
↪portal address to take effect.
```

Delete the portal address for an iSCSI target service:

```
Target> iscsi target portal del
ACCESS Target SUCCESS V-288-0 Portal IP deleted successfully
```

Create an iSCSI target:

```
Target> iscsi target create iqn.2017-02.com.veritas:target03
ACCESS Target SUCCESS V-288-0 Target iqn.2017-02.com.veritas:target03 created␣
↪successfully.
```

Destroy a specific iSCSI target:

```
Target> iscsi target destroy iqn.2001-04.com.veritas:target2
ACCESS Target SUCCESS V-288-0 iSCSI target deletion for iqn.2001-04.com.
↪veritas:target2 was completed successfully.
```

List all or specific iSCSI target:

```
Target> iscsi target list
Target Name                      Target Id   Parent Node   Accounts
==============================   ==========  ============  ====================
iqn.2001-04.com.veritas:target1  1           VA_01         ['user1', 'user2']
iqn.2001-04.com.veritas:target2  2           VA_01         []
```

Check the status of a specific iSCSI target:

```
Target> iscsi target status iqn.2017-10.com.veritas:target11
ACCESS Target SUCCESS V-288-0 Status of target iqn.2017-10.com.veritas:target11 is␣
↪ONLINE
```

Create a LUN with the specified name, LUN ID and size:

```
Target> iscsi lun create lun3 3 iqn.2017-03.com.veritas:target1 200G
ACCESS Target SUCCESS V-288-0 Lun lun3 created successfully and added to target iqn.
↪2017-03.com.veritas:target1
```

Destroy a specific LUN:

```
Target> iscsi lun destroy 4 iqn.2017-02.com.veritas:target2
ACCESS Target SUCCESS V-288-0 Lun deletion for lun id 4 was successful.
```

List the details of all the LUNs present on the system for all the targets:

```
Target> iscsi lun list
Target Name                        Lun Name   Lun ID   Lun Size
================================   =========  =======  =========
iqn.2017-02.com.veritas:target2  lun_1      21       400G
iqn.2017-02.com.veritas:target2  lun_2      22       200G
iqn.2017-02.com.veritas:target2  lun_3      23       300G
```

List the details of the LUNs which are mapped to a specified target:

```
Target> iscsi lun list iqn.2017-02.com.veritas:target1
Lun Name   Lun ID   Lun Size   Store
=========  =======  =========  ==========
lun1       12       2M         /vx/fs1/12
lun2       13       2M         /vx/fs1/13
lun3       14       2k         /vx/fs1/14
```

Map an iSCSI initiator to a specific iSCSI target:

```
Target> iscsi target map add iqn.2001-04.com.veritas:26064.01 iqn.2017-02.com.
↪veritas:target1
ACCESS Target SUCCESS V-288-0 Add initiator iqn.2001-04.com.veritas:26064.01 mapping␣
↪from target iqn.2017-02.com.veritas:target1 successfully
```

Remove the mapping of iSCSI initiator from specific iSCSI target:

```
Target> iscsi target map delete iqn.2001-04.com.veritas:26064.01 iqn.2017-02.com.
↪veritas:target1
ACCESS Target SUCCESS V-288-0 Remove initiator iqn.2001-04.com.veritas:26064.01␣
↪mapping from target iqn.2017-02.com.veritas:target1 successfully
```

Map the store to specific iSCSI target:

```
Target> iscsi target store add fs3 iqn.2017-02.com.veritas:target1
ACCESS Target SUCCESS V-288-0 FS fs3 is added to iSCSI target iqn.2017-02.com.
↪veritas:target1.
```

Remove the mapping of store from specific iSCSI target:

```
Target> iscsi target store delete fs3 iqn.2017-02.com.veritas:target1
ACCESS Target SUCCESS V-288-0 FS fs3 is deleted from iSCSI target iqn.2017-02.com.
↪veritas:target1.
```

Create an incoming user and bind the account to a specified, existing iSCSI target:

```
Target> iscsi target auth incominguser add robin robin  iqn.2017-02.com.
↪veritas:target03
Input password for robin
Enter Password:
Re-enter Password:
ACCESS Target SUCCESS V-288-0 Add user robin successfully.
```

Remove an incoming user and unbind the account from its corresponding iSCSI target:

```
Target> iscsi target auth incominguser delete robin target1
ACCESS Target SUCCESS V-288-0 Delete user robin successfully.
```

20.1.5 SEE ALSO

target_iscsi_lun(1), target_iscsi_service(1), target_iscsi_target(1)

20.2 iscsi lun

20.2.1 SYNOPSIS

`iscsi lun destroy [lunid] [target-name]`

`iscsi lun create` *lun-name lunid target-name size*

`iscsi lun destroy` *lunid target-name*

`iscsi lun list` [*targetname*]

20.2.2 DESCRIPTION

The `iscsi lun` commands are used to perform LUN-related operations.

20.2.3 OPTIONS

`iscsi lun create` *lun-name lunid target-name size* Create a LUN with the specified name, LUN ID and size on the specified file system.

`iscsi lun destroy` *lunid target-name* Destroy a specific LUN.

`iscsi lun list` [*targetname*] List details of all the LUNs present in all or a specific target.

20.2.4 EXAMPLES

Create a LUN with the specified name, LUN ID and size:

```
Target> iscsi lun create lun3 3 iqn.2017-03.com.veritas:target1 200G
ACCESS Target SUCCESS V-288-0 Lun lun3 created successfully and added to target iqn.
↪2017-03.com.veritas:target1
```

Destroy a specific LUN:

```
Target> iscsi lun destroy 4 iqn.2017-02.com.veritas:target2
ACCESS Target SUCCESS V-288-0 Lun deletion for lun id 4 was successful.
```

List the details of all the LUNs present on the system for all the targets:

```
Target> iscsi lun list
Target Name                      Lun Name   Lun ID   Lun Size
==============================   =========  =======  =========
iqn.2017-02.com.veritas:target2  lun_1      21       400G
iqn.2017-02.com.veritas:target2  lun_2      22       200G
iqn.2017-02.com.veritas:target2  lun_3      23       300G
```

List the details of the LUNs which are mapped to a specified target:

```
Target> iscsi lun list iqn.2017-02.com.veritas:target1
Lun Name   Lun ID   Lun Size   Store
=========  =======  =========  ==========
lun1       12       2M         /vx/fs1/12
lun2       13       2M         /vx/fs1/13
lun3       14       2k         /vx/fs1/14
```

20.2.5 SEE ALSO

target(1), target_iscsi_service(1), target_iscsi_target(1)

20.3 iscsi target service

20.3.1 SYNOPSIS

`iscsi service` start|stop|status

20.3.2 OPTIONS

`iscsi service` start|stop|status Start, stop or display the status of the iSCSI target service.

20.3.3 EXAMPLES

Start the iSCSI target service. If the iSCSI target service is already started, the start command clears the faults, if any, and then tries to start the iSCSI target service:

```
Target> iscsi service start
ACCESS Target SUCCESS V-288-0 iSCSI Target service started
```

Stop the iSCSI target service:

```
Target> iscsi service stop
ACCESS Target SUCCESS V-288-0 iSCSI Target service stopped
```

Check the status of iSCSI target service:

```
Target> iscsi service status
Portal IP :- 192.168.10.33
Node   Status
=====  ========
VA_01  OFFLINE
VA_02  OFFLINE
```

20.3.4 SEE ALSO

target(1), target_iscsi_lun(1), target_iscsi_target(1)

20.4 iscsi target

20.4.1 SYNOPSIS

`iscsi target portal add` *portal-ip*

`iscsi target portal del`

`iscsi target create` *targetname*

`iscsi target destroy` *targetname*

`iscsi target list` [*targetname*]

`iscsi target status` *target-name*

`iscsi target store add` *fs-name target-name*

`iscsi target store delete` *fs-name target-name*

`iscsi target map add` *initiator-name target-name*

`iscsi target map delete` *initiator-name target-name*

`iscsi target auth incominguser add` *username target-name*

`iscsi target auth incominguser delete` *username target-name*

20.4.2 DESCRIPTION

The `iscsi target` command creates, manages and destroys targets on Veritas Access. You can perform the following functions on an iSCSI target:

1. Addition and deletion of targets using the `iscsi target` commands
2. Associating file system as backing store for a target using the `iscsi target store` commands
3. Map and un-map initiators using the `iscsi target map` commands
4. Addition and deletion of users to set up CHAP authentication

20.4.3 OPTIONS

`iscsi target portal add` ***portal-ip*** Add the portal address for an iSCSI target service.

`iscsi target portal del` Delete the portal address for an iSCSI target service.

`iscsi target create` ***targetname*** Create an iSCSI target.

`iscsi target destroy` ***targetname*** Destroy a specific iSCSI target.

`iscsi target list` **[*targetname*]** List all or specific iSCSI targets.

`iscsi target status` ***target-name*** Check the status of a specific iSCSI target.

`iscsi target store add` ***fs-name target-name*** Map file system with specified iSCSI target.

`iscsi target store delete` ***fs-name target-name*** Remove the file system from a specified iSCSI target.

`iscsi target map add` ***initiator-name target-name*** Map the iSCSI initiator with specified iSCSI target.

`iscsi target map delete` ***initiator-name target-name*** Remove the iSCSI initiator mapped to a specified iSCSI target.

iscsi target auth incominguser add *username target-name* Create an incoming user and bind the account to a specified, existing iSCSI target.

iscsi target auth incominguser delete *username target-name* Remove an incoming user and unbind the account from its corresponding iSCSI target.

20.4.4 EXAMPLES

Add the portal address for an iSCSI target service:

```
Target> iscsi target portal add 192.168.10.32
ACCESS Target SUCCESS V-288-0 Portal add successful. Start the iSCSI service for the␣
↪portal address to take effect.
```

Delete the portal address for an iSCSI target service:

```
Target> iscsi target portal del
ACCESS Target SUCCESS V-288-0 Portal IP deleted successfully
```

Create an iSCSI target:

```
Target> iscsi target create iqn.2017-02.com.veritas:target03
ACCESS Target SUCCESS V-288-0 Target iqn.2017-02.com.veritas:target03 created␣
↪successfully.
```

Destroy a specific iSCSI target:

```
Target> iscsi target destroy iqn.2001-04.com.veritas:target2
ACCESS Target SUCCESS V-288-0 iSCSI target deletion for iqn.2001-04.com.
↪veritas:target2 was completed successfully.
```

List all or specific iSCSI target:

```
Target> iscsi target list
Target Name                      Target Id   Parent Node   Accounts
==============================   ==========  ============  ====================
iqn.2001-04.com.veritas:target1  1           VA_01         ['user1', 'user2']
iqn.2001-04.com.veritas:target2  2           VA_01         []
```

Check the status of a specific iSCSI target:

```
Target> iscsi target status iqn.2017-10.com.veritas:target11
ACCESS Target SUCCESS V-288-0 Status of target iqn.2017-10.com.veritas:target11 is␣
↪ONLINE
```

Map an iSCSI initiator to a specific iSCSI target:

```
Target> iscsi target map add iqn.2001-04.com.veritas:26064.01 iqn.2017-02.com.
↪veritas:target1
ACCESS Target SUCCESS V-288-0 Add initiator iqn.2001-04.com.veritas:26064.01 mapping␣
↪from target iqn.2017-02.com.veritas:target1 successfully
```

Remove the mapping of iSCSI initiator from specific iSCSI target:

```
Target> iscsi target map delete iqn.2001-04.com.veritas:26064.01 iqn.2017-02.com.
↪veritas:target1
ACCESS Target SUCCESS V-288-0 Remove initiator iqn.2001-04.com.veritas:26064.01␣
↪mapping from target iqn.2017-02.com.veritas:target1 successfully
```

Map the store to specific iSCSI target:

```
Target> iscsi target store add fs3 iqn.2017-02.com.veritas:target1
ACCESS Target SUCCESS V-288-0 FS fs3 is added to iSCSI target iqn.2017-02.com.
↪veritas:target1.
```

Remove the mapping of store from specific iSCSI target:

```
Target> iscsi target store delete fs3 iqn.2017-02.com.veritas:target1
ACCESS Target SUCCESS V-288-0 FS fs3 is deleted from iSCSI target iqn.2017-02.com.
↪veritas:target1.
```

Create an incoming user and bind the account to a specified, existing iSCSI target:

```
Target> iscsi target auth incominguser add robin robin  iqn.2017-02.com.
↪veritas:target03
Input password for robin
Enter Password:
Re-enter Password:
ACCESS Target SUCCESS V-288-0 Add user robin successfully.
```

Remove an incoming user and unbind the account from its corresponding iSCSI target:

```
Target> iscsi target auth incominguser delete robin target1
ACCESS Target SUCCESS V-288-0 Delete user robin successfully.
```

20.4.5 SEE ALSO

target(1), target_iscsi_lun(1), target_iscsi_service(1)

21 Upgrade Commands

21.1 upgrade

21.1.1 SYNOPSIS

`version`

`history`

`repository`

`get` *URL*

`install` *version*

21.1.2 DESCRIPTION

The `upgrade` commands install patches to the Veritas Access software. The software is installed on all the nodes. The `upgrade version` command also shows the current software version.

21.1.3 OPTIONS

`version` Displays the current running version of Veritas Access software.

`history` Displays the upgrade history of Veritas Access software.

`repository` Display all releases that are available in the repository.

`get` *URL* Downloads the release and puts it in the repository.

`install` *version* Upgrade the cluster to the specified version.

21.1.4 EXAMPLES

Displays the current running version of Veritas Access software:

```
    Upgrade> version
ACCESS 7.1.0.0     (Fri Mar 11 12:49:05 2016)
```

Displays the upgrade history of Veritas Access software:

```
    Upgrade> history
ACCESS 7.0.0.0     (Thu November  8 00:00:00 CST 2015),  Installed on Fri Sep 19␣
↪00:00:00 CST 2015
ACCESS 7.1.0.0     (Mon Jul  7 00:00:00 CST 2015),  Installed on Fri Sep 19 00:00:00␣
↪CST 2015
```

Download a patch from an HTTP server with authentication:

```
    Upgrade> get http://admin@docserver.veritas.com/patches/ACCESS.tar.gz
Enter password for user 'admin': **********
ACCESS upgrade NOTICE V-288-0 ACCESS patch downloaded and stored in repository.
```

Display all releases that are available in the repository:

```
    Upgrade> repository
ACCESS VERSION|  INSTALLED(y/n)|  UPGRADABLE(y/n)
7.1.0.1        y                n
7.1.0.2        n                y
7.2.0.0        n                y
```

Upgrade the cluster to the specified version:

```
    Upgrade> install 6.0.0.2
Veritas Veritas Access 6.0.0.2 Upgrade Program
```

21.1.5 SEE ALSO

get(1), history(1), install(1), repository(1), version(1)

21.2 get

21.2.1 SYNOPSIS

`get` *URL*

21.2.2 DESCRIPTION

The upgrade `get` *URL* command downloads the release and puts it in the repository.

21.2.3 OPTIONS

`get` *URL* The command downloads the patch release from the URL and put it in the repository. It also validates the patch release downloaded and shows error if the downloaded release is not valid. The URL is the URL location from where you can download the software patch. The URL supports HTTP, FTP, and SCP protocols for download. The user name and password for the HTTP and FTP protocols are also supported.

21.2.4 EXAMPLES

Downloads the release and puts it in the repository:

```
Upgrade> get http://admin@docserver.veritas.com/patches/ACCESS.tar.gz
Enter password for user 'admin': **********
ACCESS upgrade NOTICE V-288-0 ACCESS patch downloaded and stored in repository.

Upgrade> get scp://root@10.198.89.178:/patches/ACCESS.tar.gz
Enter password for user 'root': **********
ACCESS upgrade NOTICE V-288-0 ACCESS patch downloaded and stored in repository.
```

21.2.5 SEE ALSO

patch(1)

21.3 history

21.3.1 SYNOPSIS

`history`

21.3.2 DESCRIPTION

The upgrade `history` command displays the upgrade history of Veritas Access software.

21.3.3 OPTIONS

`history` Displays the upgrade history of Veritas Access software, including the ACCESS software release date, the ACCESS software upgrade history. It also shows error messages if any of the nodes do not have matching software versions installed.

21.3.4 EXAMPLES

Displays the upgrade history of Veritas Access software:

```
    Upgrade> history
ACCESS 7.0.0.0     (Thu November  8 00:00:00 CST 2015),  Installed on Fri Sep 19␣
↪00:00:00 CST 2015
ACCESS 7.1.0.0     (Mon Jul  7 00:00:00 CST 2015),  Installed on Fri Sep 19 00:00:00␣
↪CST 2015
```

21.3.5 SEE ALSO

patch(1)

21.4 install

21.4.1 SYNOPSIS

`install` *version*

21.4.2 DESCRIPTION

The upgrade `install` *version* command upgrade the cluster to the specified version.

21.4.3 OPTIONS

`install` ***version*** The ACCESS upgrade install commands perform a direct upgrade or a guided rolling upgrade on all of the clusters nodes based on the patch type. Before you perform an upgrade, ensure that all the cluster nodes are in RUNNING state and have the same Veritas Access software version.

21.4.4 EXAMPLES

Upgrade the cluster to the specified version:

```
    Upgrade> install 7.1.0.2
Veritas Veritas Access 7.1.0.2 Upgrade Program
```

21.4.5 SEE ALSO

patch(1)

21.5 repository

21.5.1 SYNOPSIS

`repository`

21.5.2 DESCRIPTION

The upgrade `repository` command displays all releases that are available in the repository.

21.5.3 OPTIONS

`repository` Displays all releases that are available in the repository. It also shows if a patch release in the repository is installed or not, or the patch release can be used to upgrade the cluster.

21.5.4 EXAMPLES

Display all releases that are available in the repository:

```
    Upgrade> repository
ACCESS VERSION|  INSTALLED(y/n)|  UPGRADABLE(y/n)
7.1.0.1        y                n
7.1.0.2        n                y
7.2.0.0        n                y
```

21.5.5 SEE ALSO

patch(1)

21.6 version

21.6.1 SYNOPSIS

`version`

21.6.2 DESCRIPTION

The upgrade `version` command displays the current running version of Veritas Access software.

21.6.3 OPTIONS

`version` Displays the current running version of Veritas Access software, including the ACCESS software release date. It also shows error messages if any of the nodes do not have matching software versions installed.

21.6.4 EXAMPLES

Displays the current running version of Veritas Access software:

```
    Upgrade> version
ACCESS 7.2.1.0     (Fri Mar 11 12:49:05 2016)
```

21.6.5 SEE ALSO

patch(1)